



meshIQ Platform Core Services

User's Guide

Version 11

Document Title: meshIQ Platform Core Services User Guide

Document Release Date: July 2024

Document Number: CS-USR11.000

Product Release: 11.0.1 (JRE 11)

Published by:

Research & Development
meshIQ
88 Sunnyside Blvd, Suite 101
Plainview, NY 11803

Copyright © 2010–2024. All rights reserved. No part of the contents of this document may be produced or transmitted in any form, or by any means without the written permission of meshIQ.

Confidentiality Statement: The information within this media is proprietary in nature and is the sole property of meshIQ. All products and information developed by meshIQ are intended for limited distribution to authorized meshIQ employees, licensed clients, and authorized users. This information (including software, electronic and printed media) is not to be copied or distributed in any form without the expressed written permission from meshIQ.

Acknowledgements: The following terms are trademarks of meshIQ in the United States or other countries or both: AutoPilot, AutoPilot M6, M6 Web Server, M6 Web Console, M6 for WMQ, MQControl, Navigator, XRay.

The following terms are trademarks of the IBM Corporation in the United States or other countries or both: IBM, MQ, WebSphere MQ, WIN-OS/2, AS/400, OS/2, DB2, Informix, AIX, and z/OS.

InstallAnywhere is a trademark or registered trademark of Flexera Software, Inc.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>), including Derby Database Server. The Jakarta Project" and "Tomcat" and the associated logos are registered trademarks of the Apache Software Foundation.

Intel, Pentium and Intel486 are trademarks or registered trademarks of Intel Corporation in the United States, or other countries, or both.

Microsoft, Windows, Windows NT, Windows XP, the Windows logos, Microsoft SQL Server, and Microsoft Visual SourceSafe are registered trademarks of the Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Mac, Mac OS, and Macintosh are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

"Linux" and the Linux Logos are registered trademarks of Linus Torvalds, the original author of the Linux kernel. All other titles, applications, products, and so forth are copyrighted and/or trademarked by their respective authors.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other company, product, and service names may be trademarks or service marks of others.

Table of Contents

CHAPTER 1: INTRODUCTION	1
1.1 USE OF THIS DOCUMENT	1
1.2 GUIDE ORGANIZATION	1
1.3 HISTORY OF THIS DOCUMENT	2
1.3.1 <i>User Feedback</i>	2
1.4 RELATED DOCUMENTS.....	2
1.5 RELEASE NOTES	2
1.6 INTENDED AUDIENCE.....	2
1.7 SYSTEM REQUIREMENTS	2
1.8 TECHNICAL SUPPORT.....	2
1.8.1 <i>The Resource Center</i>	3
1.9 CONVENTIONS	3
CHAPTER 2: ABOUT MESHIQ PLATFORM CORE SERVICES	4
2.1 INTRODUCING CORE SERVICES	4
2.2 TERMS, CONCEPTS, AND ARCHITECTURE	6
2.2.1 <i>Concepts and Architecture</i>	6
2.2.2 <i>meshIQ Services</i>	7
2.2.3 <i>CEP Servers</i>	7
2.2.4 <i>System Services</i>	7
2.2.5 <i>Monitoring Services</i>	8
2.2.6 <i>N-Tier Service Oriented Architecture</i>	13
2.2.7 <i>Typical Core Services Installation</i>	14
2.3 NATIVE SECURITY MODEL	16
2.3.1 <i>Security Requirements</i>	16
2.3.2 <i>Accounts and Passwords</i>	17
2.3.3 <i>Permissions and Ownership</i>	18
2.3.4 <i>Access Control</i>	18
2.4 LDAP INTEGRATION.....	19
2.4.1 <i>LDAP Requirements</i>	19
2.4.2 <i>Configuring Domain Server for LDAP</i>	20
2.4.3 <i>Configuring Domain Server for LDAP (Windows Active Directory)</i>	27
2.4.4 <i>Accounts</i>	27
2.4.5 <i>Permissions and Ownership</i>	28
2.5 USER INTERFACE	28
2.5.1 <i>User Console</i>	29
CHAPTER 3: GETTING STARTED.....	30
3.1 BASICS	30
3.1.1 <i>Launching the meshIQ Platform in a Windows Environment</i>	30
3.1.2 <i>Launching the meshIQ Platform from the Command Prompt</i>	32
3.1.3 <i>Stopping Core Services</i>	36
3.1.4 <i>Starting and Stopping Core Services on UNIX</i>	37
3.1.5 <i>Start and Stop Services Using apnet</i>	38
3.1.6 <i>Getting Started with the User Console</i>	40
3.2 INSTALLATION MANAGER.....	50
CHAPTER 4: ADMINISTERING CORE SERVICES.....	52

4.1	LICENSING.....	52
4.1.1	<i>Obtaining Licenses</i>	52
4.1.2	<i>Checking the License Expiration Date</i>	52
4.1.3	<i>Checking License Status</i>	53
4.2	MANAGING USERS AND GROUPS	54
4.2.1	<i>User Manager</i>	54
4.2.2	<i>Adding Users</i>	59
4.2.3	<i>Adding Groups</i>	63
4.2.4	<i>Removing Users</i>	65
4.2.5	<i>Removing Groups</i>	65
4.2.6	<i>Changing Passwords</i>	66
4.2.7	<i>Importing Users and Groups</i>	68
4.3	SERVICE AND ACCOUNT PERMISSIONS	69
4.3.1	<i>Service Permission Mask</i>	69
4.3.2	<i>Account Permission Mask</i>	69
4.4	CEP SERVERS	71
4.4.1	<i>Performance Guidelines</i>	71
4.4.2	<i>CEP Server Communication</i>	72
4.4.3	<i>Using CEP Servers</i>	73
4.5	EXPERTS	75
4.5.1	<i>Built-in Experts</i>	75
4.5.2	<i>Deploying Experts</i>	82
4.5.3	<i>Deploying Process Wrapper</i>	83
4.5.4	<i>Configuring Process Wrapper</i>	84
4.5.5	<i>Deploying File Monitors</i>	98
4.5.6	<i>Deploying IBM MQ Experts</i>	106
4.6	MANAGERS	106
4.6.1	<i>Built-in Managers</i>	107
4.6.2	<i>Deploying Managers</i>	109
4.6.3	<i>Manager Configuration</i>	111
4.7	POLICIES	119
4.7.1	<i>Deploying Policies</i>	120
4.7.2	<i>Configuring Policies</i>	125
4.7.3	<i>Using Managers for Displaying Policy Information</i>	128
4.7.4	<i>Policy Profiler</i>	129
4.7.5	<i>Health and Load Balancing Policy</i>	131
4.8	BUSINESS VIEWS	136
4.8.1	<i>Business View Deployment Cycle</i>	136
4.8.2	<i>Exploring and Managing Business Views</i>	137
4.8.3	<i>Business View Sensors</i>	139
4.9	ALERTS, NOTIFICATIONS AND RULES.....	143
4.9.1	<i>Configuring Alerts</i>	143
4.9.2	<i>Defining Sensor Rules</i>	146
4.9.3	<i>Dynamic Sensors</i>	159
4.9.4	<i>Automated Actions</i>	172
4.9.5	<i>Managing Sensors</i>	175
4.9.6	<i>Business Process</i>	179
4.9.7	<i>Displaying Sensors Graphically</i>	180
4.9.8	<i>Setting User Actions</i>	183
4.9.9	<i>State Change Delay</i>	186
4.9.10	<i>Specifying Maintenance Schedule</i>	188

4.9.11	<i>Ignoring Facts</i>	191
4.9.12	<i>Maintaining Sensor History</i>	191
4.9.13	<i>Sensor Performance Counters</i>	196
4.9.14	<i>Documenting Sensor Information</i>	197
4.10	REAL-TIME MONITORING.....	197
4.10.1	<i>Monitoring Business Views</i>	198
4.11	EVENT LOGGING	211
4.11.1	<i>Viewing Event Logs</i>	213
4.11.2	<i>Event Log Options</i>	215
4.12	PERFORMANCE MONITORING.....	220
4.12.1	<i>Creating Chart Profiles</i>	220
4.12.2	<i>Monitoring Facts using Performance Monitor</i>	222
4.13	SEARCHING THE MESHIQ DOMAIN.....	224
4.14	FORWARDING EVENTS TO OVO FROM BUSINESS VIEWS	225
4.14.1	<i>Issue opcmmsg with a Message Text</i>	226
4.14.2	<i>Examples of Sending Alerts</i>	227
4.15	STREAMING DATA	229
4.15.1	<i>Stream Data</i>	229
4.15.2	<i>Stream Facts</i>	230
4.15.3	<i>Logging Policies</i>	234
CHAPTER 5: CUSTOMIZING CORE SERVICES		236
5.1	OVERVIEW	236
5.2	JAVA RUNTIME—LAX FILE CUSTOMIZATION.....	238
5.2.1	<i>Using Server JVM</i>	238
5.3	SERVER RUNTIME – PROPERTY FILES	239
5.4	KEY PERFORMANCE PROPERTIES.....	248
CHAPTER 6: TROUBLESHOOTING TECHNIQUES		249
6.1	OVERVIEW	249
6.2	EVENT LOGS	249
6.2.1	<i>System Logs</i>	249
6.2.2	<i>User-defined Logs</i>	250
6.3	SERVICE ACTIVITY	251
6.4	EVENT VIEWER	253
6.5	GENERATING NRD FILES	254
CHAPTER 7: TROUBLESHOOTING PROCEDURES		255
7.1	PROBLEMS AND SOLUTIONS.....	255
7.1.1	<i>Installation</i>	255
7.1.2	<i>Business Views</i>	256
7.1.3	<i>Facts</i>	257
7.1.4	<i>CEP Server</i>	257
7.2	FAQs.....	259
7.2.1	<i>How to Stop/Start Core Services</i>	259
7.2.2	<i>What does the icon (gray ball with green refresh arrows) represent?</i>	259
APPENDIX A: REFERENCES		260
A.1	DOCUMENTATION	260
A.2	IBM DOCUMENTATION	260
A.3	HP OPENVIEW DOCUMENTATION	260

A.4	JAVA™ 2 J2SE™ FOR HP-UX INFORMATION LIBRARY	260
A.5	APACHE TOMCAT DOCUMENTATION REFERENCES	260
A.6	ORACLE ONLINE DOCUMENTATION HTTP://OTN.ORACLE.COM/DOCUMENTATION/CONTENT.HTML.....	260
A.7	TRU64 UNIX ONLINE DOCUMENTATION AND REFERENCES	260
APPENDIX B: CONVENTIONS.....		261
B.1	TYPOGRAPHICAL CONVENTIONS.....	261
APPENDIX C: COMMAND REFERENCE.....		262
C.1	PKGMAN – PRODUCT MAINTENANCE.....	262
C.2	APNET – CONTROL UTILITY.....	264
C.3	APFACT – FACT PUBLISHER.....	266
C.4	ATPNODE – CEP SERVER.....	268
C.5	ATPNAMES – DOMAIN SERVER.....	269
APPENDIX D: BEST PRACTICES		271
D.1	NAMING CONVENTIONS	271
D.2	GUIDELINES FOR BUILDING POLICIES.....	271
D.2.1	<i>Modeling</i>	271
D.2.2	<i>Deployment</i>	272
D.2.3	<i>Usage - Reducing Rule and Processing Delays</i>	272
D.3	KEY PERFORMANCE INDICATORS	273
D.3.1	<i>System Calendar</i>	273
D.3.2	<i>Java Memory</i>	274
D.3.3	<i>Database Logging</i>	275
D.3.4	<i>Sensor Performance</i>	276
D.3.5	<i>Sensor Runtime</i>	277
D.3.6	<i>Session</i>	278
D.3.7	<i>Topic</i>	279
APPENDIX E: REQUIRED LINUX PLATFORM CONFIGURATIONS		280
APPENDIX F: DASHBOARD DATABASE SCHEMA		281
APPENDIX G: DERIVED METRICS		283
APPENDIX I: SOFTWARE ACKNOWLEDGEMENTS		287
I.1	OPEN-SOURCE COMPONENTS	287
I.2	OPEN-SOURCE COMPONENTS LICENSED UNDER THE APACHE LICENSE VERSION 2.0.....	287
I.2.1	<i>Terms of the Apache license version 2.0</i>	287
I.3	OPEN-SOURCE COMPONENTS LICENSED UNDER THE BSD 3-CLAUSE	290
I.3.1	<i>Terms of the BSD 3-Clause</i>	290
I.4	ECLIPSE DISTRIBUTION LICENSE V.1.0	292
I.4.1	<i>Terms of the EDL v1.0 license</i>	292
I.5	OPEN-SOURCE COMPONENTS LICENSED UNDER THE EPL 1.0 (ECLIPSE PUBLIC LICENSE 1.0).....	292
I.5.1	<i>Terms of the EPL 1.0</i>	293
I.6	OPEN-SOURCE COMPONENTS LICENSED UNDER THE EPL 2.0 (ECLIPSE PUBLIC LICENSE 2.0).....	296
I.6.1	<i>Terms of the EPL 2.0</i>	297
I.7	OPEN-SOURCE COMPONENTS LICENSED UNDER THE ISC LICENSE	301
I.7.1	<i>Terms of the ISC license</i>	301
I.8	OPEN-SOURCE COMPONENTS LICENSED UNDER THE LGPL	302
I.8.1	<i>Terms of the LGPL</i>	302
I.9	OPEN-SOURCE COMPONENTS LICENSED UNDER THE MIT LICENSE	309

<i>I.9.1</i> <i>Terms of the MIT license</i>	312
I.10 OPEN-SOURCE COMPONENTS LICENSED UNDER THE OFL 1.1	313
<i>I.10.1</i> <i>Terms of the OFL 1.1</i>	313
I.11 OPEN-SOURCE COMPONENTS LICENSED UNDER THE ZLIB LICENSE	315
<i>I.11.1</i> <i>Terms of the Zlib license</i>	315
INDEX	316

Chapter 1: Introduction

Welcome to the *meshIQ Platform Core Services User's Guide*. This guide describes the management platform, terms, concepts, architecture, as well as use and administration of meshIQ Platform core services, servers, and components.



Service updates are non-cumulative service packs, and usually require a certain level of a service pack installed.

NOTE

1.1 Use of this Document

This guide is intended for anyone administering or using a meshIQ Platform product including Core Services, Data Services, Manage, Track, or a combination of these. The guide assumes that the user is familiar with general system and application management concepts, networking, and TCP/IP.

1.2 Guide Organization

[Chapter 1:](#) Introductory information relevant to the document and the meshIQ Platform.

[Chapter 2:](#) Contains a brief description of meshIQ Platform Core Services architecture, terms, and concepts.

[Chapter 3:](#) Provides instructions for operation and deployment of the meshIQ Platform Core Services.

[Chapter 4:](#) Contains task-oriented information and instructions for administering Core Services.

[Chapter 5:](#) Describes customization options of various components.

[Chapter 6:](#) Describes general troubleshooting techniques.

[Chapter 7:](#) Gives solutions to some common problems.

[Appendix A:](#) Provides detailed list of reference information required for the installation of Core Services.

[Appendix B:](#) Contains typographical conventions used in meshIQ documents.

[Appendix C:](#) Defines the meshIQ Platform Core Services command line interface.

[Appendix D:](#) Contains Core Services Best Practices.

[Appendix E:](#) Defines required Linux Platform Configurations.

[Appendix F:](#) Provides Dashboard Database Schema.

[Appendix G:](#) Provides a list of derived metrics.

[Index:](#) Contains an alphanumeric cross-reference of all topics and subjects of importance.

1.3 History of This Document

Table 1-1. Document History

Release Date	Document Number	Product Version	Summary
July 2024	CS-USR11.000	11.0.1	Initial platform release

1.3.1 User Feedback

meshIQ encourages all users of meshIQ Platform Core Services to submit comments, suggestions, corrections, and recommendations for improvement for all meshIQ documentation. Please send your comments via mail or email. Send messages to: support@meshiq.com. You will receive a written response, along with status of any proposed change, update, or correction.

1.4 Related Documents

Complete listings of documents related to meshIQ Platform Core Services can be found in [Appendix A](#).

1.5 Release Notes

See `README.HTM` file on installation media and root installation directory.

1.6 Intended Audience

This document is intended for personnel installing, customizing, and using Core Services. The user who installs the product should be familiar with:

- Java Run Time Environment 8 (JRE 8) or later for Service Updates prior to 34; JRE 11 and later for Service Updates 34 and later.
- Basic understanding of TCP/IP

1.7 System Requirements

All software and hardware requirements are defined in *the meshIQ Platform Core Services Installation Guide*.

1.8 Technical Support

If you need technical support, you can contact meshIQ by telephone or by email. To contact technical support by telephone, call **800-963-9822 ext. 1**. If you are calling from outside the United States, dial **001-516-801-2100**. To contact mySupport by email, send a message to mysupport@meshiq.com. To access the meshIQ automated mySupport system (user id and password required), go to: <https://mysupport.meshiq.com/>. Contact your local meshIQ Platform Administrator for further information.

1.8.1 The Resource Center

The Resource Center is where meshIQ users solve problems, exchange ideas, and learn best practices from peers and meshIQ staffers. This online community is a service provided by meshIQ Support. Our experts often participate in these discussion groups to share their advice, but these groups are intended as a peer-to-peer resource.

The Resource Center also provides access to downloads, updates, documentation, support articles, product news, and a lot more. Registration and access are free to all users. We encourage everyone using meshIQ products to join the [meshIQ Resource Center](#).

1.9 Conventions

Refer to [Appendix B](#) for conventions used in this guide.

Chapter 2: About meshIQ Platform Core Services

2.1 Introducing Core Services

The meshIQ Platform is designed to monitor and control distributed IT services such as application servers, middleware, user applications, workflow engines, brokers, Service Oriented Architecture (SOA) and Enterprise Service Bus (ESB) based applications and their impact on business services. It reduces time and effort required to monitor IT services end-to-end without the need for custom development, complex configuration, and customization. The meshIQ Platform is based on SOA and can run on a variety of platforms. It employs a policy-based approach to monitoring across heterogeneous platforms and applications.

(Complex Event Processor) CEP-based Event Stream Processing and Correlation engine is at the core of the meshIQ Platform technology. This engine allows aggregation, sorting, filtering, merging, and joining of various events and metric streams in real time using a wizard driven GUI interface.

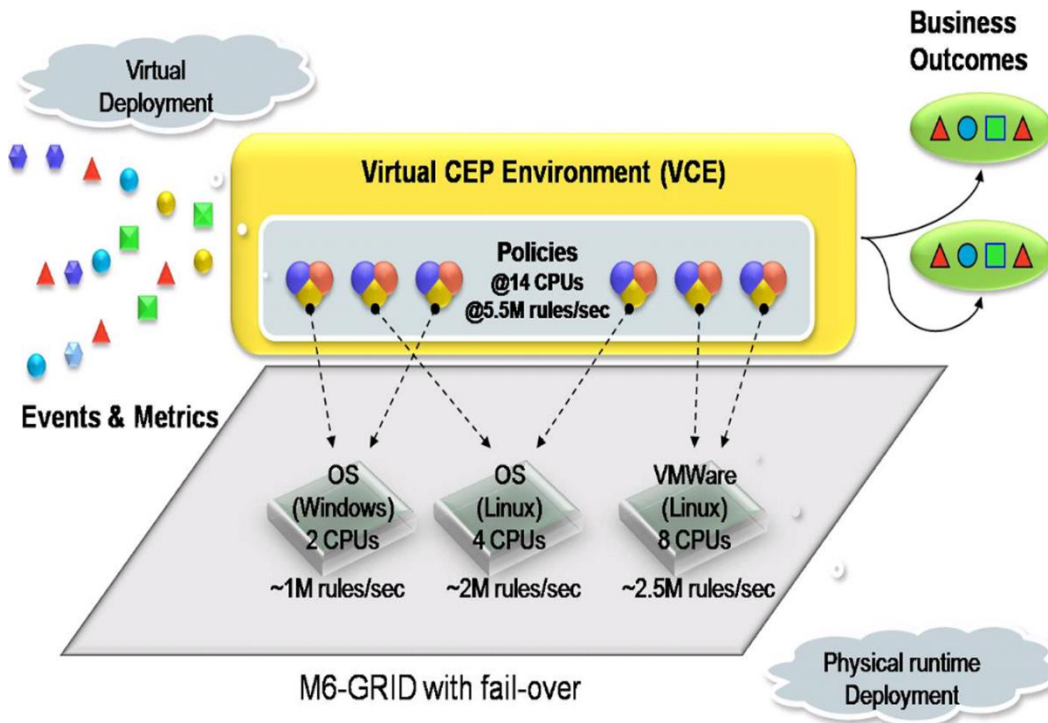


Figure 2-1. Virtual CEP

CEP instances can be replicated or auto-failed over within minutes or even seconds using grid configuration and can be consolidated or distributed on one or more hardware or Virtual Machines (VMs) -- such as VMWare or XEN. The replacement hardware/operating system does not have to be identical.

CEP has fault, performance, and security isolation. Each CEP instance is independent of the others. A crash or malfunction in one instance does not affect another instance. Its hardware includes CPU and memory isolation.

The entire state of the CEP is stored in a single deployment file. The entire CEP configuration can be transferred easily regardless of operating system, architecture, or processor.

The Virtual CEP Environment (VCE) does the following:

- Reduces time to install and configure a new monitoring domain from days/weeks to minutes
- Reduces the time to move a CEP instance to a new server or VM from days to minutes with uninterrupted operation.
- The GRID configuration provides automatic CEP instances provisioning in case one or more are offline for maintenance or any other reason.
- CEP instances can simply be allocated on the fly without service interruptions.
- Resource allocations can be made dynamically.
- Policies can be reassigned and redeployed within seconds to any CEP instance.

Whether in the cloud or on premises, the meshIQ Platform provides observability across the integration mesh. It is highly effective in environments with multiple mission-critical applications that run using a variety of technologies, from enterprise-level systems to homegrown applications.

meshIQ Platform Core Services provides several types of monitoring services: *experts*, *managers*, *policies*, and *business views*. Experts are data collection components, which reside on CEP servers. Managers are distributed software components equipped with policies (components that proactively monitor and automate). Business views are top-level visual automation tools that can be deployed as and serve as policies.

meshIQ Platform Core Services uses a collaboration model and collects all related system and application metrics (facts) from managed resources. It correlates the facts via intelligent business views, which can be shared among teams within the organization.

The meshIQ Platform provides security based on the mutual authentication between a client and a server, or between one server and another, before a network connection is opened between them.

It is important to note that understanding the platform's terms, concepts, and architecture is essential for administrators. Some of the terms in this document may not be commonly used in the systems and application management space. Understanding the meshIQ platform's architecture and organization becomes critical, since it employs a blend of n -tier and network architecture, which is contrary to traditional 3-tier application and system management platforms. This architecture allows it to scale beyond the limitations of standard 3-tier models and provide users with unmatched flexibility, scalability, and performance. Please review [section 2.2](#), Terms, Concepts, and Architecture, thoroughly before deploying the meshIQ Platform full scale.

2.2 Terms, Concepts, and Architecture

2.2.1 Concepts and Architecture

The meshIQ Platform employs agent and agent-less technology with real-time, distributed event stream processing and correlation engine. It utilizes advanced data collection techniques and collects application component-level performance, availability, and operational metrics from any of the managed applications as well as user-defined components. *As a result, the meshIQ Platform is highly scalable; capable of processing millions of rules per second and monitoring small-, medium-, and large-scale infrastructures.* Capacity can be added on the fly by creating instances of CEP servers (event stream engines).

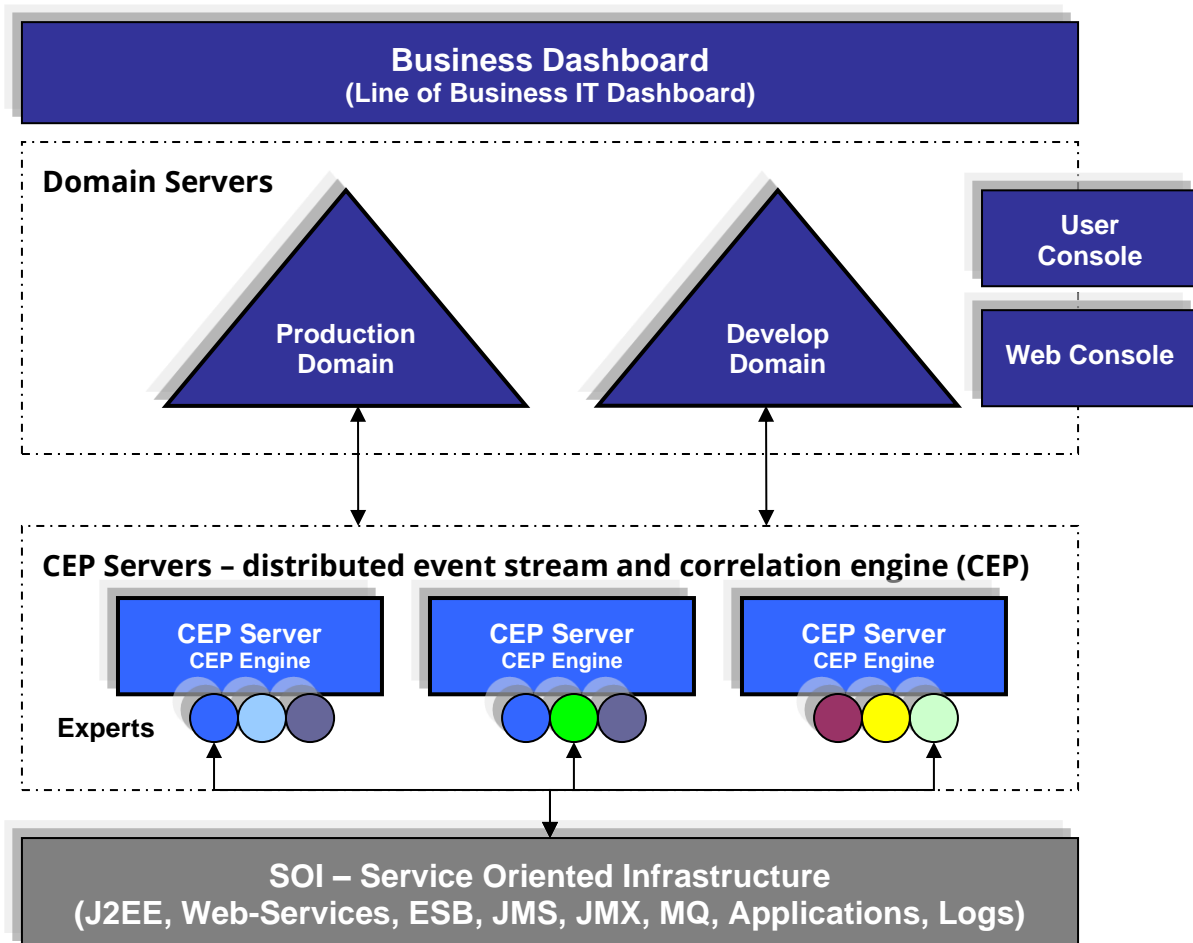


Figure 2-2. meshIQ Platform Core Services Architecture

2.2.2 meshIQ Services

meshIQ services are monitoring agents, rules, policies, and automation scripts that monitor and act on a managed environment. An example would be an expert that monitors the performance and health of a Windows platform.

**NOTE**

Do not run the Core Services in a mixed JRE environment. Ensure that all services, Domain Server, CEP Servers and the User Console use a consistent JRE level. Prior to Service Update 34, JRE 8 or higher is supported. For Service Updates 34 and later, JRE 11 and later are supported.

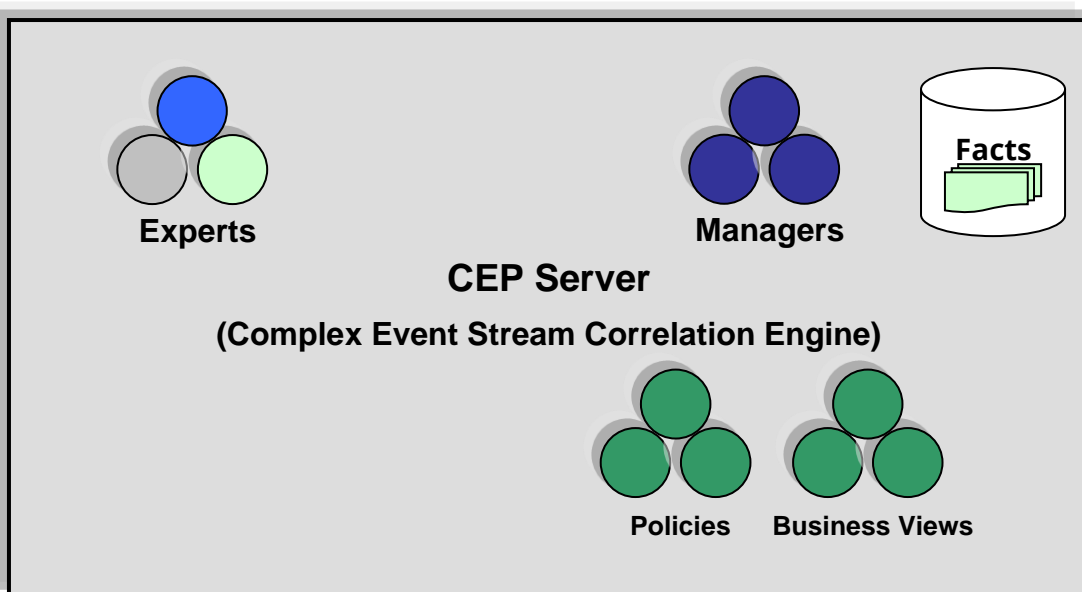


Figure 2-3. Core Services Components

2.2.3 CEP Servers

CEP servers are a collection of containers that host management services. All CEP servers are registered within the domain server.

2.2.4 System Services

- **DOMAIN_SERVER:** Reserved CEP server, which provides centralized directory and security services.
- **SYSTEM:** There is a system folder under each CEP server.
 - Reserved and not to be modified.
 - System Services should never be deleted, moved, or modified.

2.2.4.1 Domain Server

- Domain Server is the main component of the meshIQ network.
- Domain server is a CEP server that consolidates security and directory services
- Domain Server is a primary CEP server and is capable of hosting management services
- There must be at least one domain server running on a network.

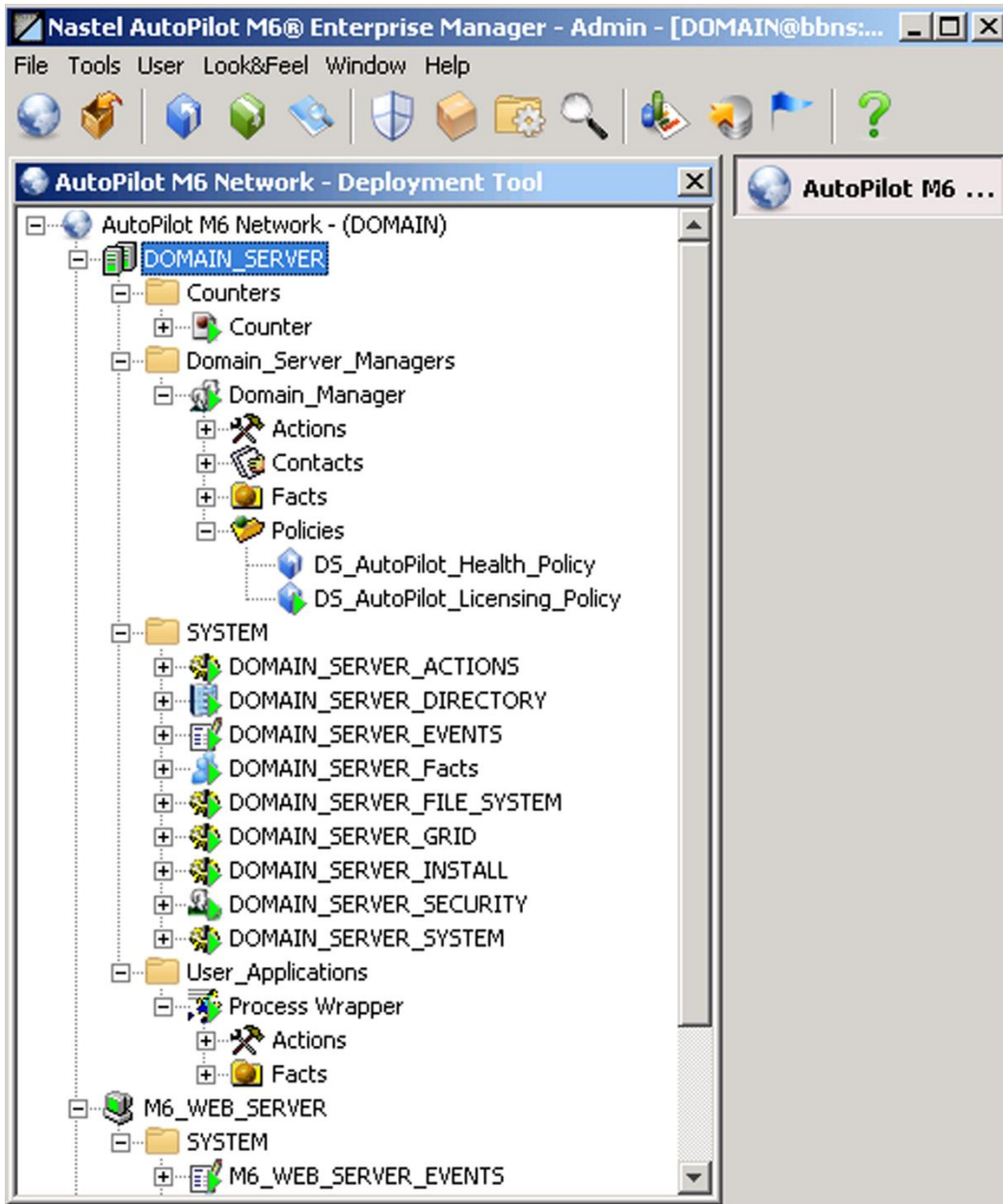


Figure 2-4. Domain Server View

2.2.5 Monitoring Services

- **Experts and Agents:** components that collect data (performance, availability etc.) from underlying applications. The primary gateway between Core Services and the target application.

- **Managers:** Managers are non-specific monitors that manage and dispatch policies and business views. Managers allow partitioning of policies and business views into groups based on purpose, security, and access control.
- **Policies:** Policies use proactive automation rules and procedures to perform actions on one or more management services on behalf of a user (for example, scheduling a policy).
- **Business Views:** Business views are a collection of rules that define a desired state of an *application* environment. Business views can be tailored to present the information in the form that best suits your needs. They represent a dependency model (tree form), where every element (sensor) describes how each application component should operate.

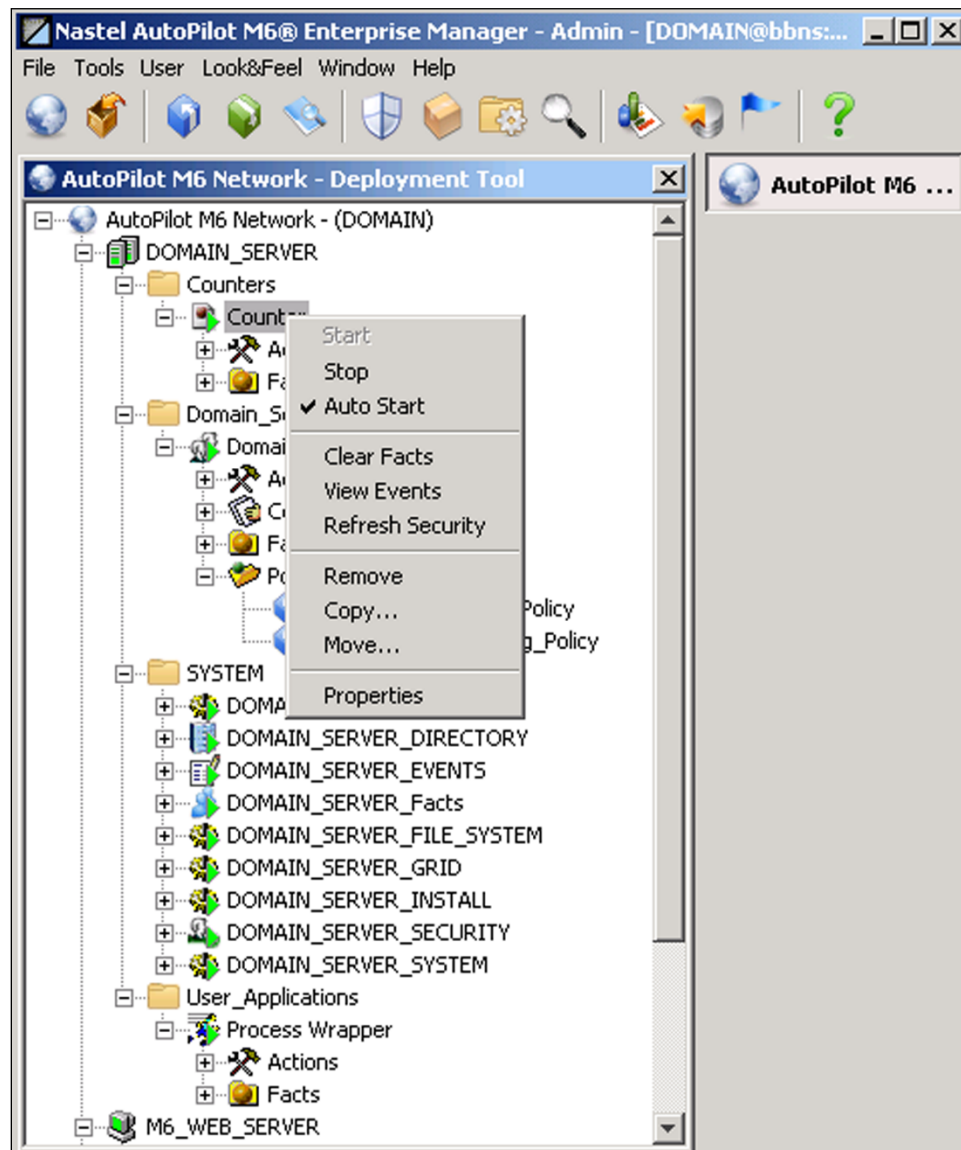


Figure 2-6. Management Services View

2.2.5.1 Experts

The official name for an application specific module is *Expert*.

- Experts monitor and control specific applications.
- Experts interface with managed applications using application-specific interfaces.
- Experts do not make judgments about the managed environment.
- Experts focus on data collection and execution of actions.

2.2.5.2 Managers

- Application independent monitoring and control agents
- Control and monitor multiple experts or managers
- Can be aggregated into *n*-tier management hierarchy
- Host and execute policies and business views

A manager has a set of policies and contacts. Policies are described in the following section. Contacts are other management services such as managers or experts that are associated with the given manager. Managers automatically subscribe to facts published by the services listed in the *Contacts* folder. These facts are routed to policies. The contact list is created dynamically and depends on the deployed policies and the services that these policies reference.

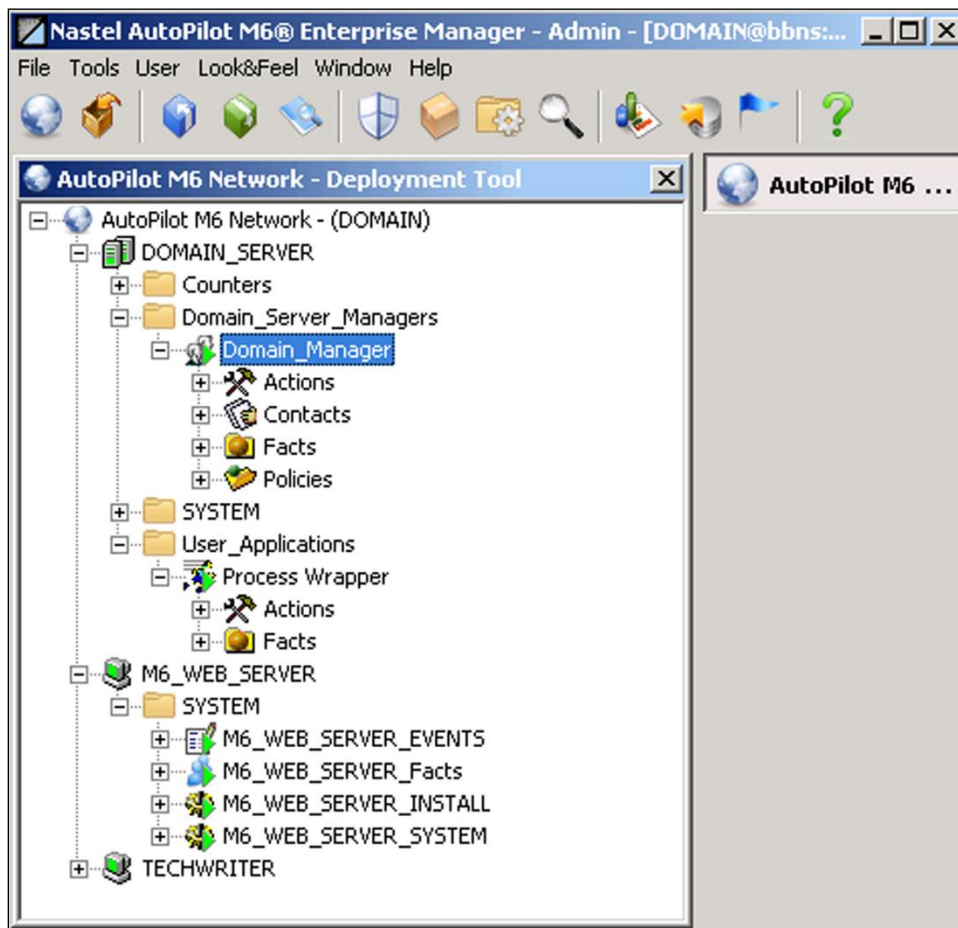


Figure 2-7. Manager View

2.2.5.3 Policies

- Perform actions on one or more management services
- Write information to database, log files, and other data stores
- Subscribe to facts and act on fact changes, time events, or other conditions
- Are usually specific to monitored applications
- Are deployed within managers (example: Alert or execute an action based on a condition/event)

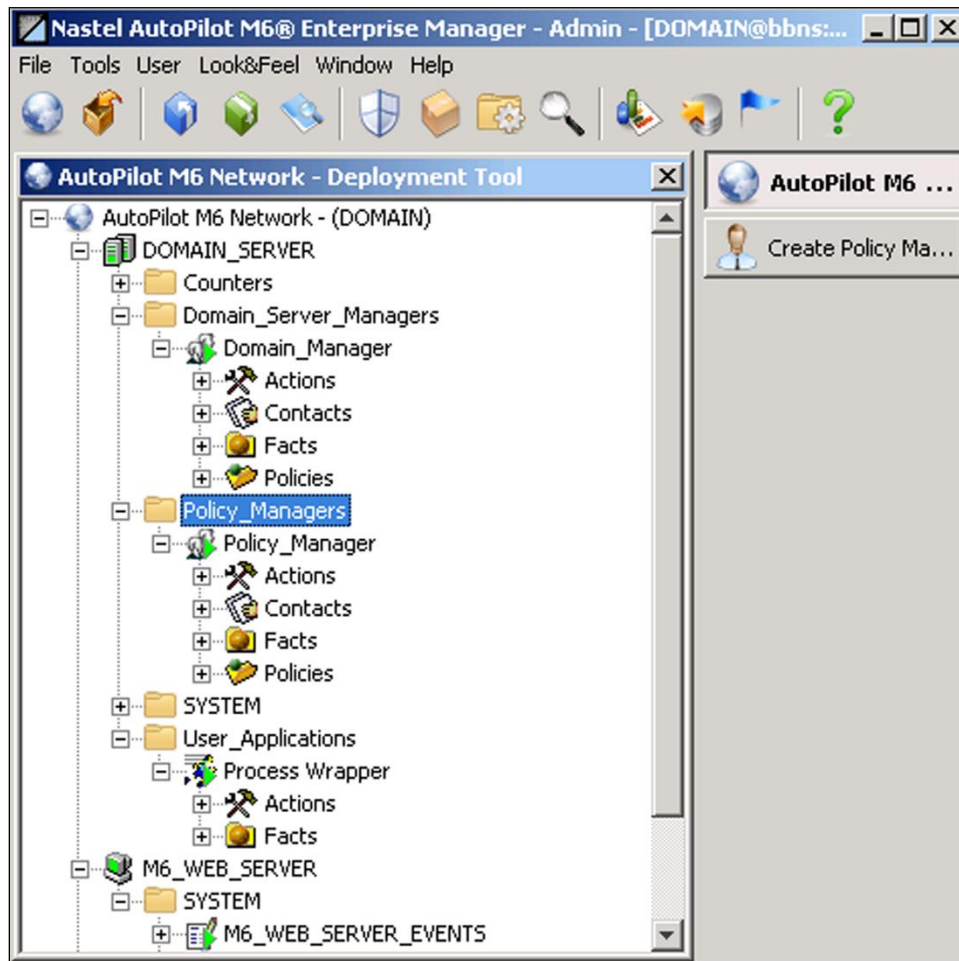


Figure 2-8. Policy View

2.2.5.4 Business Views

- Proactive user-defined policies that:
 - Correlate facts/events
 - Automate and alert
 - Generate user-defined events
 - Collect historical data for future analysis
- Business views are defined using the User Console
- Deployable as policies within any defined network of managers (running in the background)

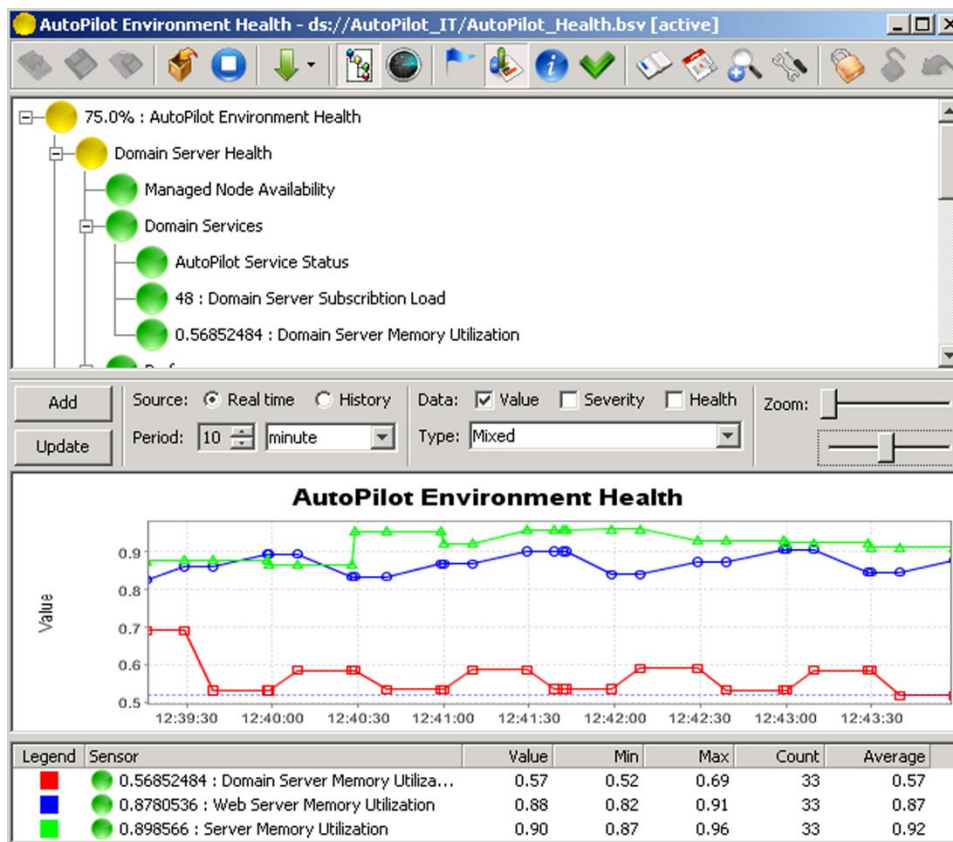


Figure 2-9. Business View Example

2.2.5.5 Core Services Facts – Metrics

- Facts are attributes or metrics published by management services about the target application, system, business process, or a system process. Examples:
 - Represents the state of an object or application
 - Performance data about applications, process, or resource
 - Metrics related to an application or business process
- Facts are variables that may change in real time
- Facts follow this format: **Variable=Value/Method** where **Variable**, **Value**, and **Method** are objects (Example: `CPUIDLE=90`)

- Fact names are broken down using “\” into groups of facts
- System metric example:
 - OS\Windows\Server1\CPU\IDLE=90
 - Applications\SAP_R3\Running=true
 - Web\MQ\Home_Page\Response=20
- Business metric example:
 - Inventory\Products\Machines=1020
 - Financial\DOW_Average=35000
 - Financial\NASDAQ_Average=16000

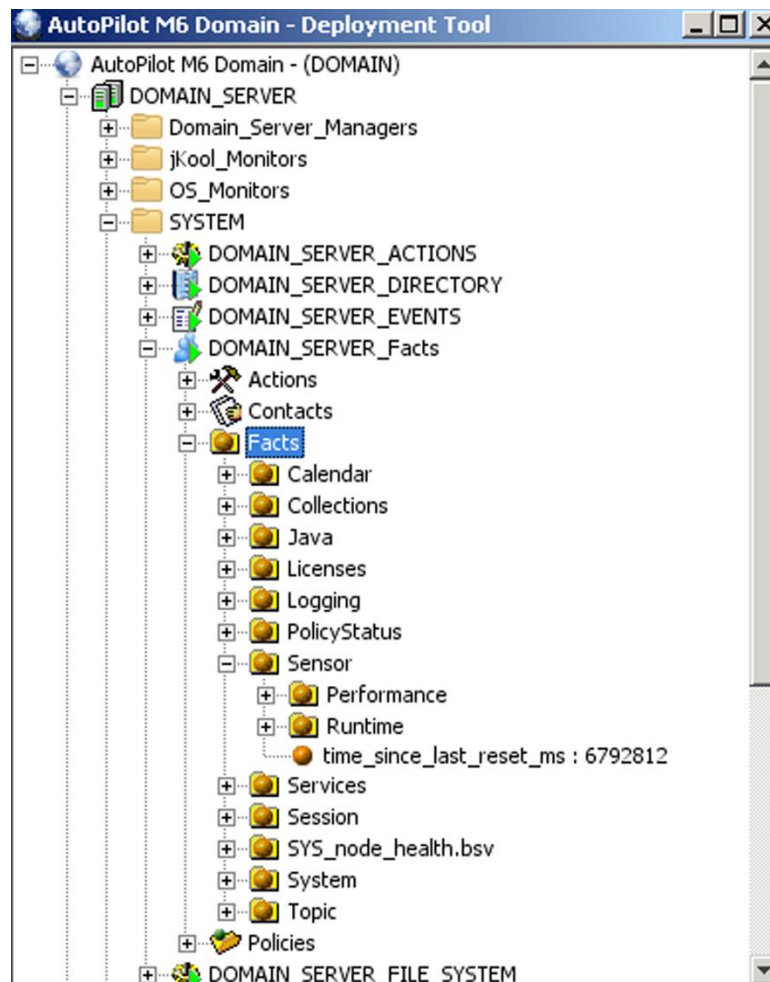


Figure 2-10. Facts View

2.2.6 N-Tier Service Oriented Architecture

The illustration below shows how Facts published by experts/agents flow into one or more subscribers. These subscribers are other management services such as managers, policies, and business views. Each management service will interpret the facts and produce:

- Other combined Facts
- Corrective or recovery actions

- Alerts and records information to database

Since CEP servers host management services, the architecture, in whole or in part, may be deployed into a single CEP server or can be split into multiple CEP servers.

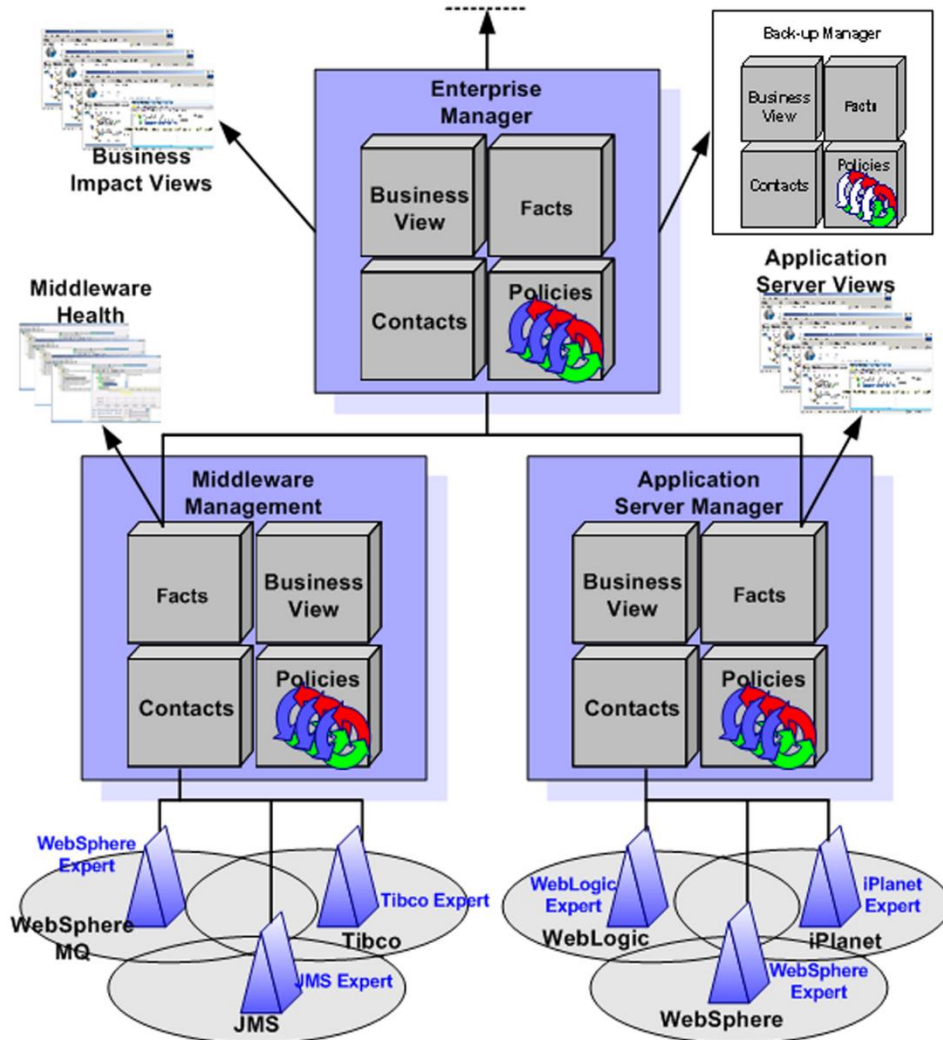


Figure 2-11. N-Tier Service Oriented Architecture

2.2.7 Typical Core Services Installation

The following represents a typical installation.

- Domain Server
 - Installed on a single machine
- CEP Server
 - Installed on every machine where monitoring is required
- User Console
 - Installed for every meshIQ administrator
 - Allows users to administer the meshIQ domain
- **Installation Layout.** The illustration below reflects the follow conditions:
 - Domain Server is installed in a Windows 10 (or later) environment. Only one domain server is required on the network.

- CEP Servers are installed on other machines and are registered with the Domain Server.
- User consoles are installed on Windows/UNIX workstations

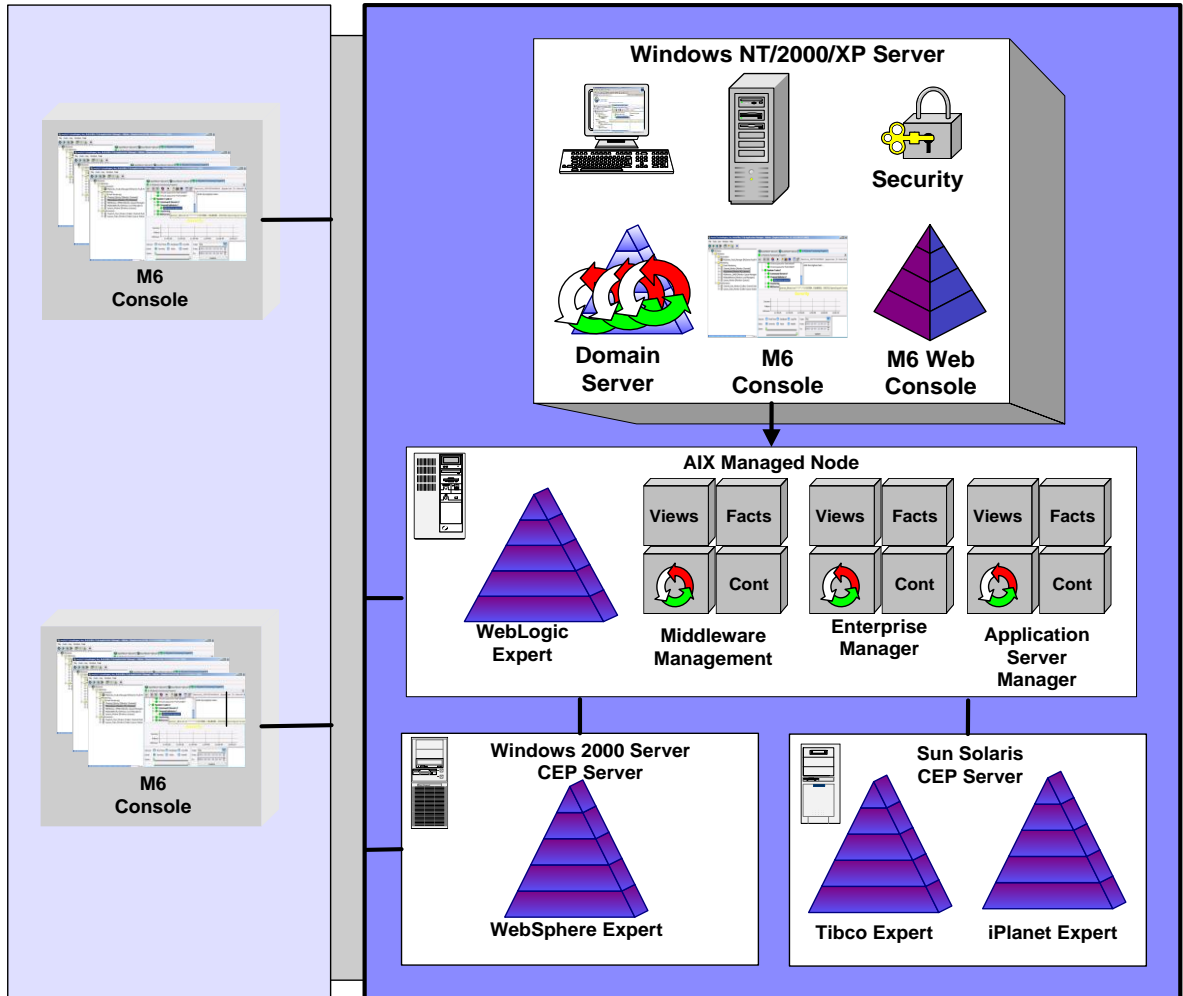


Figure 2-12. Typical Installation Layout

2.3 Native Security Model

The meshIQ Platform provides a role-based security model, where all users must be authenticated with the central trusted security authority: the Domain Server. The Domain Server assigns public and private keys for each created account, including groups. Upon successful authentication, the domain server issues a digital signature that is signed with the account's public key. The issuing Domain Server verifies the digital signature, since it is the only authority that has the account's private key. The Domain Server must run on a trusted secure host, which is guarded by sound security procedures.

All accounts are stored on the domain server in the *security.dat* file under the `[AUTOPILOT_HOME] \naming` directory. The file is stored in binary format and contains accounts, private-public keys, and encrypted passwords.



IMPORTANT!

Do not delete the *security.dat* file, but **do** back it up regularly. Loss or corruption of this file will prevent users from logging on and may cause run-time problems. This file must also be protected against modification or any malicious activity.

2.3.1 Security Requirements

The security requirements for the meshIQ Platform vary depending on the extent of use. Adjustments have to be made depending on your specific use. For example, if you need to collect statistics to a database, then relevant access is required to that database. Listed below are some common requirements:

- **Access to TCP/IP services on all systems.** The following are default listening ports (configurable) for each of the products:

Workgroup Server -	4010		
M6-WMQ Agent -	5000		
Domain Server -	2323	3000	8889
CEP Server -	2325	3005	

- **On IBM MQ Servers:** The ID that runs the agent needs the following access to IBM MQ:

Connect	Get
Create	Put
Change	Alternate User
Inquire	Set
Display	

There may be others, depending on the extent of use.

- **On MVS:** Read, Update, Alter and Control.
- **On WAS Servers:** The `pmiexpert` is used. If global security is turned *On*, the user id and password must be set in `sas.client.props`, `soap.client.props`. The

passwords can then be encoded with
[wasroot]/bin/PropFilePasswordEncoder.bat for enhanced security.

- **On HPOV SPI:** The SPI installation creates some Node Groups, Message Groups, Applications, User Profiles, and User. Authorized users must be given access to these Node and Message Groups.

2.3.2 Accounts and Passwords

User Groups

A Group is a collection of user accounts that are logically related and share the same rights. Groups may contain users or other groups. There are two default user groups: *Administrators* and *Operators*. Only a user with administrative privileges (member of *Administrators* group) can add, delete, and modify the users and groups. The *Administrators* group is a system group and cannot be deleted.

User Accounts

Your local meshIQ Platform administrator assigns user IDs and passwords. As a user, you cannot add or delete accounts, unless you are part of the *Administrators* group. meshIQ provides a *SYSTEM* account, which is reserved for system use and cannot be deleted.

Passwords

Your initial user password is issued by the system administrator, or as local policy dictates. The default password for the *Administrator* and *Admin* accounts is *admin* (case sensitive). The passwords must be changed soon after the installation.

Password and password format policies should be adopted and enforced to prevent unauthorized system access.



NOTE

Please remember your password; passwords cannot be recovered due to one-way password encryption. User names and passwords are case sensitive.

Changing Passwords

Native users must change their passwords after first logon in accordance with domain security policy. The default minimum character length is five which is set from domain server properties `server.security.password.length=5`. The changed password will be in effect the next time you log on. To change your password in the future, see your administrator.



Figure 2-13. Changing User Password

2.3.3 Permissions and Ownership

Each management service has an owner (that is, a user who deployed/created the service) and a permission mask, which applies to all services that the user creates. Each user group has Administrator-defined access and limitations based on account policies. See your meshIQ Administrator for additional information.

The security tab for each expert, manager, policy, and business view identifies the owner, whether the service inherits the permissions from the originator/owner and permission settings. The permissions are inherited from owner's account "Security" settings and subdivided into three categories:

- **Group Permission** – the mask applies to all members of the account's group, which may contain other groups.
- **Others Permission** – the mask applies to a set of users that are not part of the group membership of a specific account.
- **Permissions (Supplementary)** – the mask applies to a group or individual user. It is used to grant additional permissions. Permissions granted here are added to the permissions granted by the **Group** or **Others** settings.

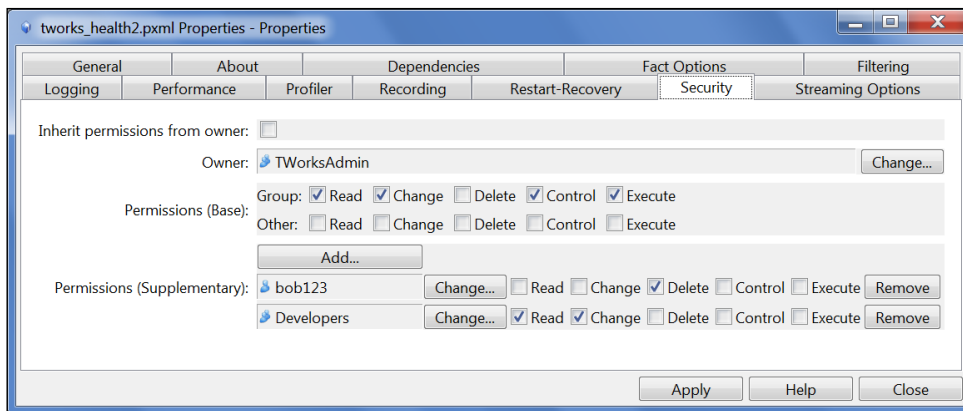


Figure 2-14. Security Tab

Changing the permission mask at the account level propagates the change to all owned services that inherit owner permissions. Propagation occurs only when the CEP server that hosts the services is restarted.

2.3.4 Access Control

Members of the *Administrators* group have full access to all non-administrator owned objects within the meshIQ Platform, even when owners deny access to all other accounts. Objects owned by members of the *Administrator* group follow standard access control procedures based on the object permission settings. By default, the following operations are reserved for members of the *Administrators* group only:

- **Domain Repository** – ability to save, change, upload business views
- **CEP Servers** – ability to stop and control CEP servers
- **Event Logs** – ability to delete/clear event log entries.
- **Security** – manage user account and security properties.
- **Deployment** – ability to deploy new experts and managers.

2.4 LDAP Integration

The LDAP authentication consists of a client-server model that provides access to the entries in a directory structure via a directory service. The entries are referenced by their globally-unique Distinguished Name (DN). In LDAP, authentication is done by “binding” to the directory service. The data required depends on the type of authentication required, but generally requires the DN for a user and the user's password.

User/group membership can be derived from the appropriate server.

2.4.1 LDAP Requirements

To utilize strictly LDAP for authentication and group membership, the only requirements are:

- The ability for the Domain Server to connect to the LDAP service port. When configuring AutoPilot Domain Server LDAP configuration, you must specify the LDAP server URL as shown in these examples:
 - LDAP://myserver.example.com
 - LDAPS://myserver.example.com
 - LDAP://myserver.example.com:389
 - LDAPS://myserver.example.com:636
 - LDAP://myserver.example.com:3268
 - LDAPS://myserver.example.com:3269

The default port for an LDAP connection is 389 and for LDAPS is 636. When you configure an LDAP connection to use port 389/636, you search for objects from this local domain controller only (replicated between domain controllers in the same domain). It has a complete set of all the attributes each object contains. Alternatively, when configuring Core Services LDAP integration in environments with multiple domains in the forest, it is often required to use the Global Catalog to return objects from all domains in the forest. The Global Catalog is a Read Only replica which contains a Partial Attribute Set (PAS) of objects within the forest, so it holds certain replicate objects from all domains. The default port for this is 3268 for LDAP and 3269 for LDAPS. When you configure the LDAP connection to use port 3268/3269, you search this Global Catalog (GC) to locate objects from any domain without having to know the domain name itself. This is often used in multi-domain forests where Core Services must pull users/groups from multiple domains. To summarize:

- Default Ports: 389 (LDAP) / 636 (LDAPS): These ports are used for requesting information from the local domain controller. LDAP requests sent to port 389/636 can be used to search for objects only within the global catalog's home domain. However, the requesting application can obtain all of the attributes for those objects.
- Default Ports: 3268 (LDAP) / 3269 (LDAPS): These ports are used for queries specifically targeted for the Global Catalog. LDAP requests sent to port 3268/3269 can be used to search for objects in the entire forest. However, only the attributes marked for replication to the Global Catalog can be returned.

- Keystore file(s) containing certificate(s) for LDAP servers (if using SSL/TLS for LDAP server communication).

2.4.2 Configuring Domain Server for LDAP

Configuring Domain Server for LDAP requires:

- Defining necessary Domain Server LDAP properties
- Obtaining/generating a certificate for the LDAP server and adding it to keystore (for using SSL/TLS).

2.4.2.1 Configure Domain Server LDAP Properties

Configuring Domain Server for LDAP authentication consists of defining the appropriate properties in `[AUTOPILOT_HOME]/naming/node.properties`. The following properties are supported. The following template definitions are provided in `node.properties` ([Table 2-1](#)):

```
;Security LDAP Configuration Properties
;
; uncomment line below to enable LDAP authentication mode (default
is false)
;property server.domain.ldap = true
;
; uncomment below to enable security trace. Enable for
troubleshooting LDAP
;property com.nastel.nfc.security.trace = true
;
; defines a space-separated list of named LDAP servers. When
authenticating users
; the servers are searched in the order specified.
;property server.domain.ldap.servers = ldapserver1
;
;LDAP Server Configuration
; Define the following properties for each named LDAP server above
; primary URL to use to connect to server
;property          server.domain.ldap.<ldapserver1>.url      =
ldap://localhost:389
;
; failover URL to user for this server is unable to connect to
primary URL - optional
;property          server.domain.ldap.<ldapserver1>.failover_url  =
ldap://localhost:10389
;
; LDAP authentication model to use (simple|DIGEST5-MD|GSSAPI),
default=simple
```

```
; - currently only simple is supported
;property server.domain.ldap.<ldapserver1>.auth = simple
;
; indicates whether or not SSL/TLS should be used to communicate
with LDAP server
; only required when using SSL/TLS without ldaps
;property server.domain.ldap.<ldapserver1>.ssl = true
;
; if using SSL/TLS, defines keystore file containing SSL/TLS
certificate for LDAP server
;property server.domain.ldap.<ldapserver1>.ssl.keystore =
c:/temp/ApacheDS-trusted.ks
;
; defines keystore file containing SSL/TLS certificate for failover
LDAP server
;property server.domain.ldap.<ldapserver1>.ssl.failover_keystore
= c:/temp/ApacheDS-trusted.ks
;
; LDAP user defined on server to use to query the server
;property server.domain.ldap.<ldapserver1>.searchuser =
uid=admin,ou=system
;
; password for LDAP user used to query server
;property server.domain.ldap.<ldapserver1>.searchpwd = secret
;
;LDAP Server User Configuration
; base (root) entry in LDAP directory where users are defined
;property server.domain.ldap.<ldapserver1>.user.base =
ou=users,dc=example,dc=com
;
; LDAP attribute whose value should be used as the user name in
AutoPilot
;property server.domain.ldap.<ldapserver1>.user.name.attr = uid
;
; LDAP attribute whose value should be used as the description
stored in AutoPilot security record - optional
;property server.domain.ldap.<ldapserver1>.user.desc.attr =
displayName
;
; LDAP class (objectclass) value that user entries must contain -
optional
;property server.domain.ldap.<ldapserver1>.user.class = person
```

```

;
;LDAP Server Group Configuration
; base (root) entry in LDAP directory where groups are defined
;property      server.domain.ldap.<ldapserver1>.group.base      =
ou=groups,dc=example,dc=com
;
; LDAP class (objectclass) value that group entries must contain -
optional
;property      server.domain.ldap.<ldapserver1>.group.class      =
groupOfUniqueNames
;
; LDAP attribute that defines group members
;property      server.domain.ldap.<ldapserver1>.group.members.attr  =
memberOf
;
; LDAP attribute whose value should be used as the group name in
AutoPilot
;property      server.domain.ldap.<ldapserver1>.group.name.attr = cn
;
; LDAP attribute whose value should be used as the description
stored in AutoPilot
; security record - optional
;property      server.domain.ldap.<ldapserver1>.group.desc.attr    =
description
;
; defines time in seconds after which a user's group membership
should be refreshed
; from LDAP server
; (0 implies never refresh - just use membership list received
during login), default=0
;property      server.domain.ldap.<ldapserver1>.group.refresh_time  =
3600

```

Table 2-1. node.properties

Property	Description
server.domain.ldap = [true false]	Set to true to enable LDAP authentication support.
server.domain.ldap.servers = <ldapserver1> [<ldapserver2>] [...]	Space-separated list of named LDAP servers to support.

The following properties (Table 2-2) are defined for each server named in `server.domain.ldap.servers`, replacing `<ldapservice1>` with the appropriate name from list.

Table 2-2. LDAP Server node.properties	
Property	Description
NOTE: Special characters are not allowed in LDAP attributes.	
<code>server.domain.ldap.<ldapservice1>.url = <URL for LDAP server></code>	Primary URL used to connect to LDAP server, e.g., <code>ldap://localhost:389</code>
<code>server.domain.ldap.<ldapservice1>.failover_url = <URL for LDAP server></code>	URL for failover server to use when this primary URL cannot be connected to. The failover LDAP server must have the exact same directory structure and entries as the server at the primary URL.
<code>server.domain.ldap.<ldapservice1>.auth = [simple DIGEST5-MD GSSAPI]</code>	LDAP authentication model to use. Default = <code>simple</code>
<code>server.domain.ldap.<ldapservice1>.ssl = [true false]</code>	Indicates whether SSL/TLS is used to communicate with LDAP server. This is only required to use SSL/TLS when the URL is not defined using "ldaps", since with ldaps, SSL/TLS is used independent of this setting.
<code>server.domain.ldap.<ldapservice1>.ssl.keystore = <path to keystore></code>	When using SSL/TLS, defines path to the keystore file containing the security certificate for the LDAP server. Only required if the certificate for LDAP server is not signed from a known Certificate Authority.
<code>server.domain.ldap.<ldapservice1>.ssl.keystore.pwd = <password for keystore></code>	Password for accessing SSL/TLS keystore
<code>server.domain.ldap.<ldapservice1>.ssl.failover_keystore = <path to keystore></code>	When using SSL/TLS, defines path to keystore file containing security certificate for the failover LDAP server. Only required if the certificate for the LDAP server is not signed from a known Certificate Authority.
<code>server.domain.ldap.<ldapservice1>.ssl.failover_keystore.pwd = <password for keystore></code>	Password for accessing SSL/TLS keystore for failover server
<code>server.domain.ldap.<ldapservice1>.searchuser = <user DN></code>	DN of a user that is defined on an LDAP server, to be used by the Domain Server to connect to LDAP in order to execute searches. This use requires read access to the subtrees containing user and group definitions, e.g., <code>ou=users,dc=example,dc=com</code>
<code>server.domain.ldap.<ldapservice1>.searchpwd = <user password></code>	Password for searchuser defined above

Table 2-2. LDAP Server node.properties

Property	Description
server.domain.ldap.<ldapserver1>.searchscope = [subtree onelevel object]	Search scope to use when looking up users and groups. One of: -subtree – search entire subtree rooted at user.base and group.base (default) -onelevel – search just the node at user.base and group.base and the level immediately below it -object – search just the node at user.base and group.base
server.domain.ldap.<ldapserver1>.user.base = <root node DN>	DN for base (root) entry in LDAP directory under which users are defined, e.g., ou=users,dc=example,dc=com
server.domain.ldap.<ldapserver1>.user.name.attr = <user LDAP attribute>	Attribute from LDAP user entries whose value should be used as the user's name in Core Services. This represents the name under which the user will log into Core Services. Default = uid
server.domain.ldap.<ldapserver1>.user.desc.attr = <user LDAP attribute>	Attribute from LDAP user entries whose value will be used as the description stored in Core Services security record for this user. This is optional and is only utilized during LDAP user import.
server.domain.ldap.<ldapserver1>.user.class = <user LDAP class>	Defines an LDAP objectclass that user entries must contain. This is optional and is only utilized during LDAP user import.
server.domain.ldap.<ldapserver1>.user.filter = <LDAP user filter>	Filter to apply when querying for users. This is optional, but when specified, must be a properly formatted LDAP filter string.
server.domain.ldap.<ldapserver1>.group.base = <root node DN>	DN for base (root) entry in LDAP directory under which groups are defined, e.g., ou=groups,dc=example,dc=com
server.domain.ldap.<ldapserver1>.group.name.attr = <group LDAP attribute>	Attribute from LDAP user entries whose value should be used as the group's name in Core Services.
server.domain.ldap.<ldapserver1>.group.desc.attr = <group LDAP attribute>	Attribute from LDAP group entries whose value will be used as the description stored in Core Services security record for this group. This is optional and is only utilized during LDAP group import.
server.domain.ldap.<ldapserver1>.group.class = <group LDAP class>	Defines and LDAP objectclass that group entries much contain. This is optional and is only utilized during LDAP group import.

Table 2-2. LDAP Server node.properties

Property	Description
server.domain.ldap.<ldapserver1>.group.filter = <LDAP group filter>	Filter to apply when querying for groups. This is optional, but when specified, must be a properly formatted LDAP filter string
server.domain.ldap.<ldapserver1>.group.members.attribute = <group LDAP attribute>	Attribute from LDAP group entries whose values contain the user members of the group.
server.domain.ldap.<ldapserver1>.group.refresh_time = <refresh interval>	Defines the interval that group membership retrieved from LDAP server is considered stale and should be retrieved from LDAP server. A value of 0 indicates that the membership list should never be refreshed. Note that this list is always retrieved when a user logs in.
server.domain.ldap.<ldapserver1>.user.mapping.attribute = <distinguishedName>	When running an LDAP query to authenticate users and identify the user groups to which they belong, meshIQ uses the mapping attribute <i>distinguishedName</i> by default.
server.domain.ldap.<ldapserver1>.group.mapping.prefix = <prefix>	<p>If you need to modify the default LDAP query by adding a prefix to the value obtained from the user mapping attribute (user.mapping.attribute), use this property. For example, the following adds the CN (common name) as a prefix:</p> <pre>server.domain.ldap.ldapserver1.group.mapping.prefix = CN=</pre>
server.domain.ldap.<ldapserver1>.group.mapping.suffix = <suffix>	<p>If you need to modify the default LDAP query by adding a suffix to the value obtained from the user mapping attribute, use this property. For example, the following adds a comma followed by the CN (common name) and two DC (domain component) values as suffixes:</p> <pre>server.domain.ldap.ldapserver1.group.mapping.suffix = ,CN=Users,DC=meshIQ,DC=com</pre>

2.4.2.2 Defining Keystore

In order for the Domain Server to communicate with LDAP service using SSL/TLS, the LDAP service must supply a trusted certificate. A trusted certificate is one that is either signed by a known Certificate Authority (e.g., Verisign), or one that exists in a store of trusted

certificates. If you already have a properly signed certificate for your LDAP service, then no further configuration is required.

On the other hand, if you do not, then you will have to obtain a certificate and store it in a keystore accessible to the Domain Server. For the LDAP service to use SSL/TLS, it must have a public/private key pair defined and must be configured to use it. How this is done is different for each LDAP service. At this point, we're going to assume that this key pair already exists. If it does not, consult the documentation for your LDAP service.

The first thing that needs to be done is to export the LDAP service certificate from the LDAP server. This can be done with the Java utility `keytool`, using options similar to the following:

```
keytool -export -keystore <keystore-file> -alias <keystore-alias>
-file ldap_serv.cer
```

You will be prompted for the keystore password. It will store the keystore in the file named `ldap_serv.cer`. The LDAP administrator will have to run this command for you.

Once you have the certificate file, you need to import it into a new or existing keystore. This should be done on the server running the Domain Server. Again, we use the `keytool` utility to import the certificate. The command below will import the certificate exported above into a keystore named `ldap_serv.ks`, creating it if it does not exist:

```
keytool -import -file ldap_serv.cer -alias <keystore-alias>
-keystore ldap_serv.ks -storepass <keystore-password>
```

If the keystore exists, then `<keystore-password>` must match the expected one.

Java Keytool Commands: If you need to check the information within a certificate, or Java keystore, use the following commands:

- Check which certificates are in a Java keystore:
`keytool -list -v -keystore keystore.jks`
- Check a particular keystore entry using an alias:
`keytool -list -v -keystore keystore.jks -alias mydomain`
- Check a stand-alone certificate:
`keytool -printcert -v -file mydomain.crt`

Now that we have our certificate in the keystore, we put the keystore file in a directory on the Domain Server system and set the property **`server.domain.ldap.<ldapserver1>.ssl.keystore`** in `node.properties` to the full path of this file.

2.4.3 Configuring Domain Server for LDAP (Windows Active Directory)

**NOTE**

The following applies to domain server installation only.

The meshIQ domain server is capable of looking up existing users within LDAP directory service (such as Windows Active Directory). To configure domain server with LDAP, do the following:

- **Configure Domain Server properties** (LDAP)-
`[AUTOPILOT_HOME]/naming/node.properties`
- **Configure JAAS:** Java Authentication and Authorization Service configuration
(`jaas.conf`)

2.4.4 Accounts

User Groups

A Group is a collection of user accounts that are logically related and share the same rights. Groups may contain users or other groups. There are two default user groups: *Administrators* and *Operators*. Only the user with administrative privilege (member of *Administrators* group) can add, delete, and modify the users and groups. The *Administrators* group is a system group and cannot be deleted.

Groups are defined as one of 2 types:

- **Native** – these groups are defined in Core Services with no assumed relationship to groups defined elsewhere (like operation system groups or LDAP groups)
- **LDAP** – these groups are assumed to match groups defined in an LDAP server. To define groups of this type, they must be imported from an LDAP server. The main use for these types of groups is to derive the group membership from the LDAP server instead of defining the memberships in Core Services .

User Accounts

Your local meshIQ Platform administrator assigns user Ids. As a user, you cannot add or delete accounts, unless you are part of the *Administrators* group. meshIQ provides a *SYSTEM* account, which is reserved for system use and cannot be deleted.

Users are defined as one of 3 types:

- **Native** – for these users, all authentication information is stored within Core Services, and when these users log in, the specified credentials are compared against these stored values.
- **LDAP** – for these users, the authentication information is stored in an LDAP server, and when these users log in, they are authenticated against the LDAP server.

User/Group Membership

Users and Groups can be associated either explicitly, where each user is specifically assigned to one or more user groups, or derived, where the user's group membership is retrieved

from an external (e.g., LDAP) server. In order to derive a user's group membership, the user must be defined as a non-Native user and the user's security record must be configured to retrieve the group membership from the external server. The explicit user/group memberships can be defined between all types of users and groups, with the following exceptions:

- For **LDAP groups**, only Native users can be explicitly added to the group. Which LDAP users are members of this group are obtained from the LDAP server.
- For **LDAP users**, they can only be explicitly added to Native groups, since again, which LDAP groups they belong to is obtained from the LDAP server.

2.4.5 Permissions and Ownership

Each management service has an owner (a user that deployed/created the service) and a permission mask, which applies to all services that the user creates. Each user group has Administrator-defined access and limitations based on account policies. See your meshIQ Platform Administrator for additional information. The security tab ([Figure 2-14](#)) for each expert, manager, policy, and business view identifies the owner, whether the service inherits the permissions from the originator/owner and permission settings. The permissions are inherited from owner's account "Security" settings and subdivided into two categories:

- **Group Permission** – the mask applies to all members of the account's group, which may contain other groups.
- **Others Permission** – the mask applies to a set of users that are not part of the group membership of a specific account.
- **Permissions (Supplementary)** – the mask applies to a group or individual user. It is used to grant additional permissions. Permissions granted here are added to the permissions granted by the **Group** or **Others** settings.

Changing the permission mask at the account level propagates the change to all owned services that inherit owner permissions. Propagation occurs only when the CEP server that hosts the services is restarted.

2.5 User Interface

The meshIQ Platform Core Services provides two types of user interfaces. The User Console is used for administration. This interface allows users to access business views that:

- Monitor collect data based on defined requirements.
- Graphically see the status of applications and impact of failures on overall environment.
- Receive alerts when failure and/or recovery occur.
- View system performance and status statistics.

Only the User Console provides users with full capability to administer the meshIQ domain and to develop and test business views.

2.5.1 User Console



IMPORTANT!

Do not run the Core Services in a mixed JRE environment. Ensure that all services, Domain Server, CEP Servers and the User Console use a consistent JRE level. Prior to Service Update 34, JRE 8 or higher is supported. For Service Updates 34 and later, JRE 11 and later are supported.

Users have full access to all operational functions within the User Console, including tools to load and create business views. The user will be able to review properties of all services, monitor facts and events. Individual access to Core Services and ownerships are as specified by your local meshIQ Platform administrator.

Chapter 3: Getting Started

3.1 Basics

The User Console is an application that gives users access to functionality and information provided by meshIQ Platform Core Services. It is a thin client (clients are any component that uses the meshIQ Platform infrastructure). As a rule, all functions and properties are visible to all users; however, based on permissions and security, the ability to change, edit properties, and create new managers, business views, and experts is subject to meshIQ domain security policy.



NOTE


If you are running the meshIQ Platform on a Linux platform, refer to Appendix D for Linux specific configurations.

3.1.1 Launching the meshIQ Platform in a Windows Environment



NOTE

The meshIQ Domain Server must be running and highly available.

1. Click Start  > meshIQ Platform > meshIQ Enterprise Manager to open the Enterprise Manager program menu.

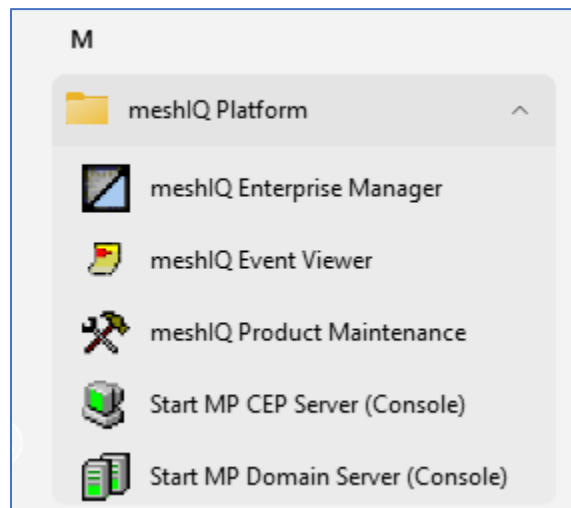


Figure 3-1. Windows Program Group for meshIQ Platform Core Services

2. The meshIQ Platform About dialog is displayed. Click **OK**.

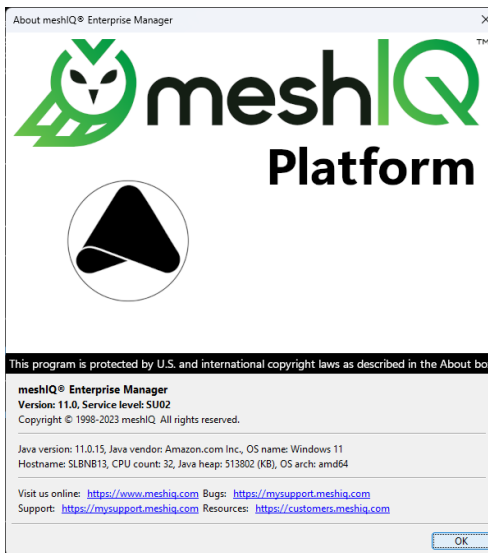



Figure 3-2. meshIQ Platform Enterprise Manager Screen

3. When the Logon screen is displayed:
 - a. If you are a first-time user, go to step 3.
 - b. Click the refresh  icon to obtain the Native domain.
 - c. Select the Security realm you will be using to log in by clicking the down arrow in Security Realm field to display the pull-down menu. The default is *DOMAIN*.
4. If you are a first-time user, enter User name **Admin** and Password **admin**. You will be prompted to change your password.
5. Enter Domain server hostname and port.
 - The Domain server field includes the domain server name and port in the format <host:port>. The default Domain Server is localhost; If running on the same server as the domain server, specify **localhost:2323**. If you want to access other domains within the meshIQ network, select the Domain Server from the menu or enter the host or IP (for example, **mpserver:2323**).
 - The Domain Port is set during installation either by the installer or by default. If you want to change it, enter a new port or select the domain port from the menu.
6. Click **OK**.

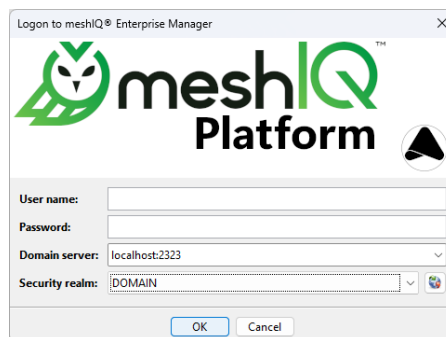


Figure 3-3. meshIQ Platform Enterprise Manager Login Screen

3.1.2 Launching the meshIQ Platform from the Command Prompt

All meshIQ components can be started from a command prompt or UNIX shell. All commands and command line options are platform independent and work on all platforms. The exact path and calling procedures are platform specific.



CAUTION!

TCP ports *must not* be shared among meshIQ Platform Core Services and any other TCP server/application on the same machine. Please refer to component specific node.properties for port configuration.

3.1.2.1 Starting Domain Server

1. From Command Prompt, `cd [AUTOPILOT_HOME]`
2. From `[AUTOPILOT_HOME]` directory type: `cd naming`.
3. From `[AUTOPILOT_HOME]\naming` type: `ATPNAMES`, to start the domain server.

```

C:\>cd nastel\autopilotm6
C:\Nastel\AutoPilotM6>cd naming
C:\Nastel\AutoPilotM6\naming>atpnames
autopilot.home = C:\Nastel\AutoPilotM6\
Parsed options: <>
Remaining args:
C:\Nastel\AutoPilotM6
Loading properties from "C:\Nastel\AutoPilotM6\global.properties"
Loading properties from "C:\Nastel\AutoPilotM6\naming\.node.prope

```

Figure 3-4. Starting a Domain Server

4. You may start the domain server in console mode, where all output is redirected to the user's console, type: `ATPNAMES -console`. All relevant data about the server and its host will be displayed. The "meshIQ [Domain] IS READY!" statement will be posted along with information and status display options.

```

2007-08-01 14:54:40 <I> -8024--deployment.Loader-Importing services from locatio
n=C:\Nastel\AutoPilotM6\naming\.import\
2007-08-01 14:54:40 <I> Started registry[C:\Nastel\AutoPilotM6\naming\.registry
.xml syncpoint thread=Thread[autopilot/Registry syncpoint for C:\Nastel\AutoPil
otM6\naming\.registry.xml,1,managed_node], timeout=3000
2007-08-01 14:54:40 <I> All system services owned by <SYSTEM>
2007-08-01 14:54:40 <I> AutoPilot M6 [Domain] is ready!
2007-08-01 14:54:40 <I> Console Commands:
2007-08-01 14:54:40 <I> 'q' -- shutdown,
2007-08-01 14:54:40 <I> 'a' -- dump all runtime environment,
2007-08-01 14:54:40 <I> 'l' -- list locally registered services,
2007-08-01 14:54:40 <I> 'p' -- list all active external processes,
2007-08-01 14:54:40 <I> 'c' -- list all active communications,
2007-08-01 14:54:40 <I> 'r' -- list all defined relations,
2007-08-01 14:54:40 <I> 's' -- list all running threads,
2007-08-01 14:54:40 <I> 'u' -- list communication usage,
2007-08-01 14:54:40 <I> 't' -- list all defined topics,
2007-08-01 14:54:40 <I> 'x' -- force memory garbage collection
2007-08-01 14:54:40 <I> 'm' -- show memory & topic utilization

```

Figure 3-5. ATPNAMES Console Mode

3.1.2.2 Starting CEP Servers



1. The Domain Server (ATPNAMES) must be running for the meshIQ Platform Core Services environment to work.
2. The following procedure is typical of starting meshIQ Platform Core Services in a Windows environment; however, it is typical of start procedures on all supported platforms.

1. From Command Prompt, type: `cd [AUTOPILOT_HOME]`
2. From `[AUTOPILOT_HOME]` directory type: `cd localhost`
3. From `[AUTOPILOT_HOME]\localhost` type: `ATPNODE -console`. Wait for node to load.

```
Command Prompt
C:\Naste1\AutoPilotM6>cd localhost
C:\Naste1\AutoPilotM6\localhost>atpnode -console
autopilot.home = C:\Naste1\AutoPilotM6\
Parsed options: {console=true}
Remaining args:
C:\Naste1\AutoPilotM6
Loading properties from "C:\Naste1\AutoPilotM6\global.properties"
Loading properties from "C:\Naste1\AutoPilotM6\localhost\.\node.properties"
start.timestamp : Wed Aug 01 15:02:27 EDT 2007
----- Operating System -----
os.name       : Windows 2000
os.version    : 5.0
os.arch       : x86
os.cpu        : 1
```

Figure 3-6. Starting CEP Server

3.1.2.3 Starting the User Console

1. From [AUTOPILOT_HOME]\mconsole type: ATPCONS. The About meshIQ Platform Enterprise Manager screen is displayed. Click **OK**.

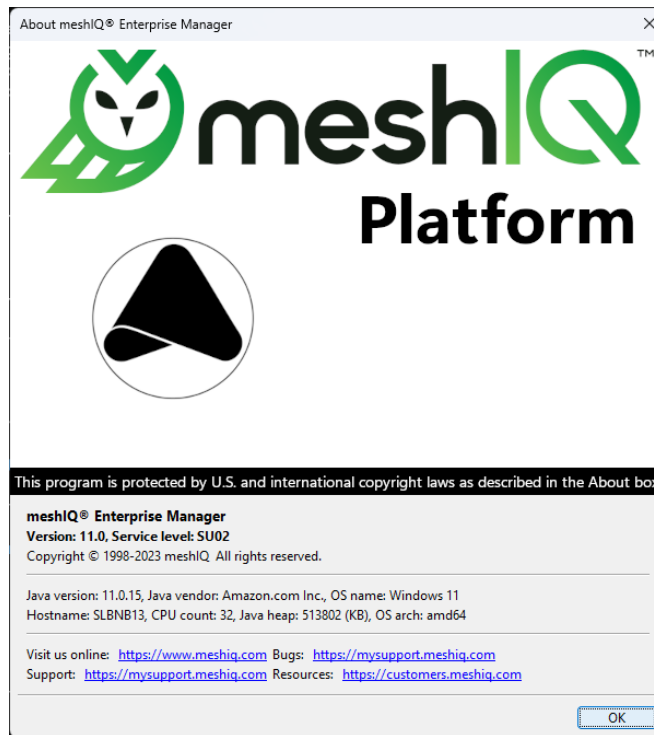



Figure 3-7. About meshIQ Platform Enterprise Manager Screen



NOTE

1. This procedure is typical of starting meshIQ Platform Core Services in a Windows environment; however, it is typical of start procedures on all supported platforms.
2. Your local meshIQ Administrator assigns Usernames and Passwords. See the System Administrator for Username and Password assignments.

2. When the Logon screen is displayed:
 - a. If you are a first-time user, go to step 3.
 - b. Click the refresh  icon to obtain the Native domain.
 - c. Select the Security Realm you will be using to log in by selecting it from the menu.
3. If you are a first-time user, enter User name **Admin** and Password **admin**. You will be prompted to change your password.
4. Enter the Domain server hostname and port.
 - The Domain server field includes the domain server name and port in the format <host:port>. The default Domain Server is localhost; If running on the same server as the domain server, specify **localhost:2323**. If you want to access other domains within the meshIQ network, select the Domain Server from the menu or enter the host or IP (for example, **mpserver:2323**).
 - The Domain Port is set during installation either by the installer or by default. If you want to change it to another, enter a new port or select the domain port from the pull-down menu.

5. Click **OK**.

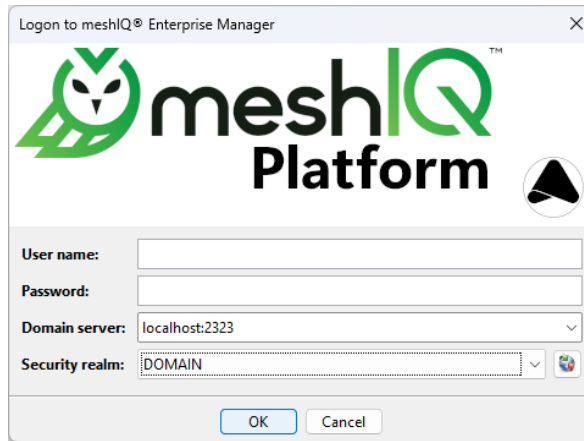


Figure 3-8. Logon to meshIQ Domain

6. When the Management Console is displayed, Enterprise Manager is running.

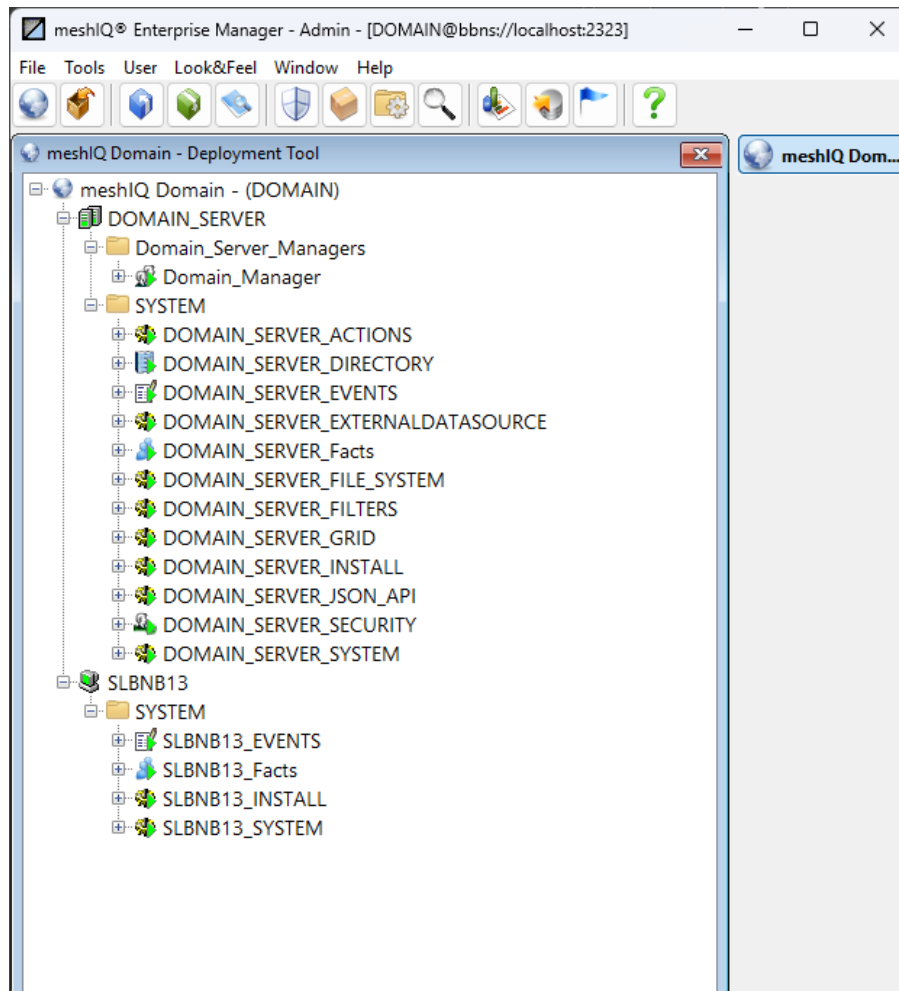


Figure 3-9. User Console

3.1.3 Stopping Core Services



There are no specific logoff procedures required in Core Services. The following is a typical example.

1. From the *Deployment Tool* screen, right-click **DOMAIN_SERVER** or the CEP server you want to stop. A sub-menu will be displayed. Click **Stop Node**. The icon of the node you stopped will switch to red and the status will be displayed as *Off-Line* (e.g., icon switches to grey) when screen is refreshed.

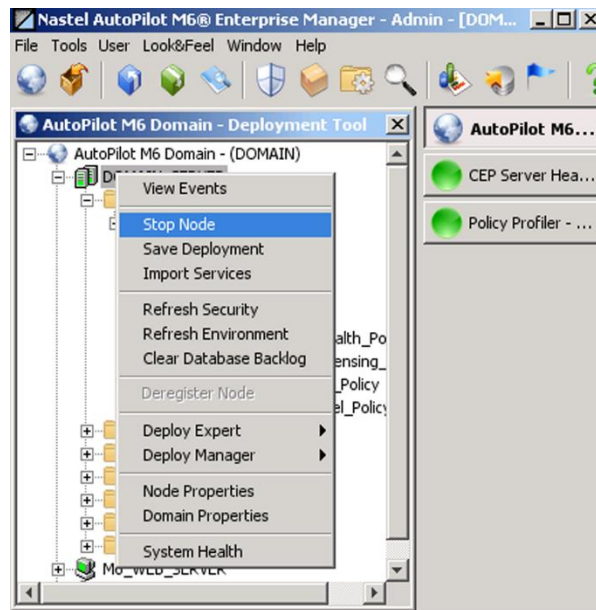


Figure 3-10. Stopping Domain Server or CEP Server (Typical)

2. Exit Consoles by selecting **File > Exit**. The screen will close; you will be logged off.

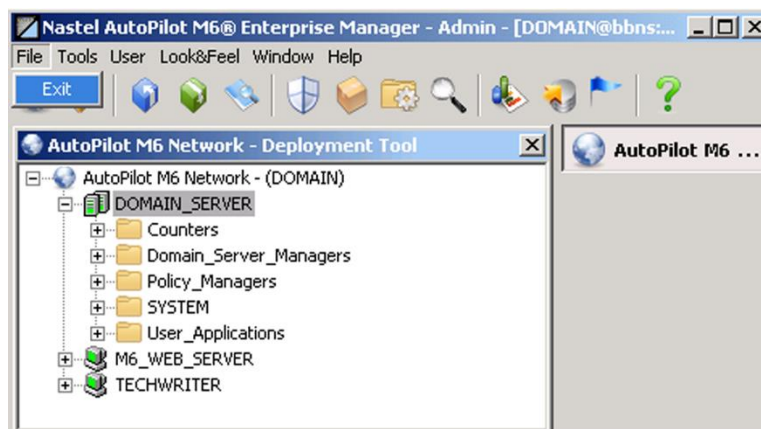


Figure 3-11. Exiting User Console, User Logoff



After exiting from the User Console, the domain server and CEP servers remain running in the background. If it is necessary to stop these components, they must be shut down manually. This can be accomplished from the command prompt or in Services (Windows).

- If the service was started with “-console” mode, you can shut down servers using the Command Prompt by typing: “q” [Enter]. Repeat this for each running service. To shut down services regardless of the location or start procedure, use the **apnet** utility.



```

C:\Nastel\AutoPilotM6>cd bin
C:\Nastel\AutoPilotM6\bin>apnet stop
AutoPilot M6 Control Utility
(c) 2000-2007 Nastel Technologies, Inc. All rights reserved.

Usage:
  apnet [Options] Command Service

Options:
  -useglb <true|false>          use settings supplied in <home>/global.pro
  -properties <default is false>
  -dsname <domain server name> domain server name <default is DOMAIN_SERV
  -server <server name>          server name <default is LOCALHOST>
  -domain <domain server host> domain server host name <default is localh
  -port <domain server port>    domain server port <default is 2323>
  -user <user name>             user name <default is SYSTEM when domain s
  -server is local)

```

Figure 3-12. Command Prompt: Domain Server

3.1.4 Starting and Stopping Core Services on UNIX

3.1.4.1 Starting Servers

- To start Core Services in a UNIX environment, use the following commands:
 - Domain Server:** `[AUTOPILOT_HOME]/naming/ATPNAMES`
 - CEP Server:** `[AUTOPILOT_HOME]/localhost/ATPNODE`
 - Web Server:** `[AUTOPILOT_HOME]/apache-tomcat/catalina.sh run`
- To start the console, start an X-Server on your workstation (if using a PC) and set the DISPLAY environment variable to your PC

Example: **ksh export DISPLAY=your_host:0**

Example: **csch setenv DISPLAY your_host:0**
- Start: `[AUTOPILOT_HOME]/mconsole/ATPCONS -console`
 where:

`[AUTOPILOT_HOME]` is the installation directory.

3.1.4.2 Stopping Servers

Stop the CEP servers, Web Server (if applicable), and domain server from within Core Services by right-clicking on the node or server, and then clicking on **Stop Node**. Stop the nodes first, then the domain server.

Alternately, you can use *apnet* or the UNIX *kill* command to stop processes ATPNAMES, ATPNODE. It is strongly recommended *not* to use the UNIX “kill -9” command.

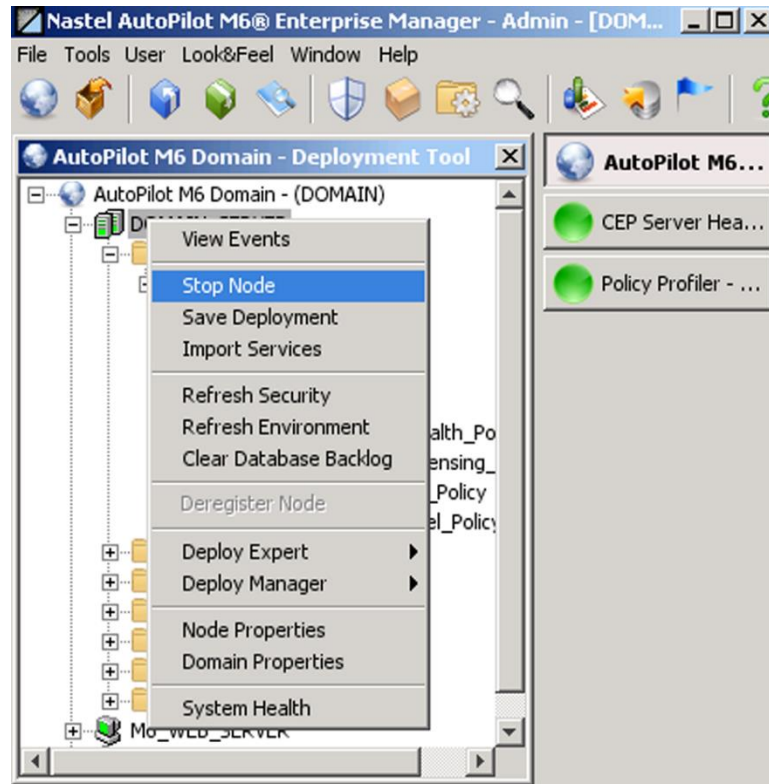


Figure 3-13. Stopping Core Services on any Platform

3.1.5 Start and Stop Services Using apnet

APNET can be used to start or stop services and stop nodes. Please refer to [C.2 APNET – Control Utility](#) for more information.

3.1.5.1 Stopping Services and CEP Servers



IMPORTANT!

CEP Servers stopped using *apnet* or the *Console* can be restarted from the command line or Windows Services. They cannot be restarted using *apnet* or *console*.

1. CD to: `[AUTOPILOT_HOME]/bin`.
2. Type: **apnet**. The *apnet options* will be displayed. The commands displayed can be used to start and stop any service or node.
3. To stop the meshIQ CEP Server, type:
`apnet -domain localhost stop <NODENAME>_SYSTEM`

4. To stop the meshIQ domain server, type:
`apnet -domain localhost stop DOMAIN_SERVER_SYSTEM`

To stop the meshIQ Web Server type:

For Windows:

1. CD to: `[AUTOPILOT_HOME]`
2. From `[AUTOPILOT_HOME]` directory type: **cd apache-tomcat**
3. From `[AUTOPILOT_HOME]\apache-tomcat` directory type: **cd bin**
4. From `[AUTOPILOT_HOME]\apache-tomcat\bin` directory type: **\shutdown.bat**

For UNIX:

1. CD to: `[AUTOPILOT_HOME]`
2. From `[AUTOPILOT_HOME]` directory type: **cd apache-tomcat**
3. From `[AUTOPILOT_HOME]\apache-tomcat` directory type: **cd bin**
4. From `[AUTOPILOT_HOME]\apache-tomcat\bin` directory type: **\shutdown.sh**

You can also use Windows Services to shut down all services.

3.1.5.2 Starting Services

The APNET command line utility can start services; however, it cannot start the meshIQ domain server, CEP Server or Web Server. These services must be started using platform-specific startup procedures. APNET can, however, start management services such as experts, managers, and policies within CEP Servers.



IMPORTANT!

CEP Servers stopped using *APNET* or *the Console* can be restarted from the command line or Windows Services. They cannot be restarted using *APNET*.

To start a management service:

```
apnet -domain localhost start Service
```

3.1.6 Getting Started with the User Console

The User Console provides your user interface for monitoring of the meshIQ domain, to create business views and monitor applications. Local security policy will dictate the specific level of access and abilities each user groups actually has been granted. All the details should be available from your Administrator. Domain security policy is managed using *User Manager* tool within the User Console.

3.1.6.1 User Console Navigation

Application Management Console:

- **Title bar:** Displays domain name and domain URL location.
- **Pull-Down Menus:** Menus provide navigation and access to standard and high-level functions within meshIQ Platform Core Services.
- **Main Toolbar:** Toolbar provides quick access to the most frequently used components.
- **Tab Selection:** Click on the tab (on the right side of the screen) to open that window. As a new window is opened, its tab is added to the right side of the screen.

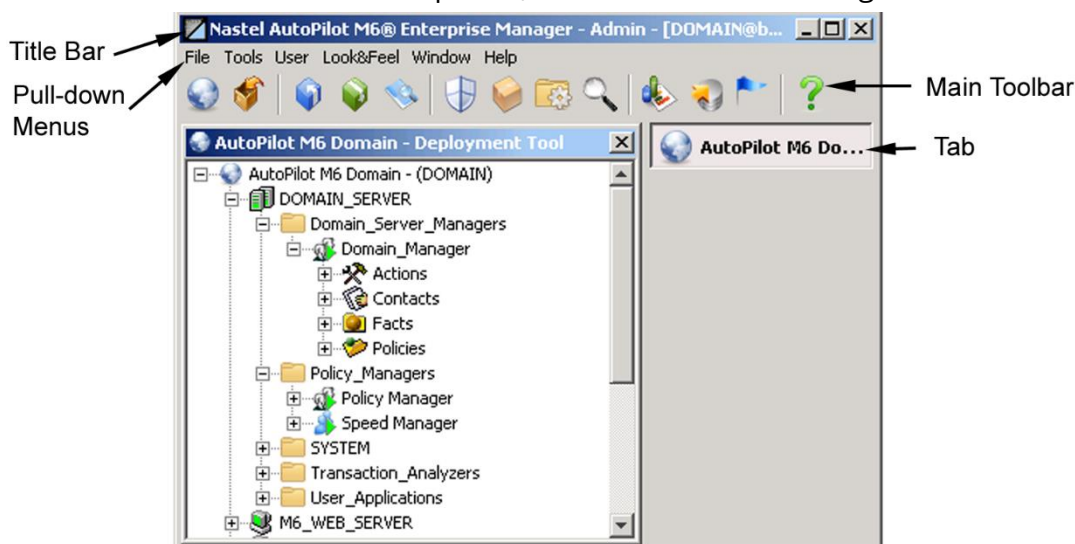


Figure 3-14. Menu and Toolbars

Pull-Down Menus

This section provides instructions and definition for functions, links, and commands related to components of the Pull-Down Menu. Pull-down menu consist of the following:

1. **File:** Click **File**, the *File* menu is displayed.
 - **Exit:** Click **Exit** to close the User Console and log off the meshIQ network.



The following menu items are only accessible if you are displaying your open views (tabs) across the top of your screen. (Select Tabs from the Windows menu.) Exit is selectable from both.

- **New:** Click **New** to display untitled window at left of screen.
- **Open:** Click **Open** to display *Open Favorite View* menu. Select the desired file. Click Open to initiate the selected favorite view. Favorites are saved with a `.cvt`

extension and are located in the

[AUTOPILOT_HOME]\mconsole\views\autopilot\custom directory.

- **Save:** Click **Save** to save the current favorite view.
- **Save As:** Click **Save As** to save a modified Favorite View. It is recommended that the built-in Favorite view remain unaltered. Once modified, rename the new view.
- **Built-In Favorite Views:** As part of the file pull-down menu the built-in favorite views are listed for quick reference. The views listed will vary according to the installation options you selected. The Customized feature and personal views will not be listed, and will require access through *Open*, above.

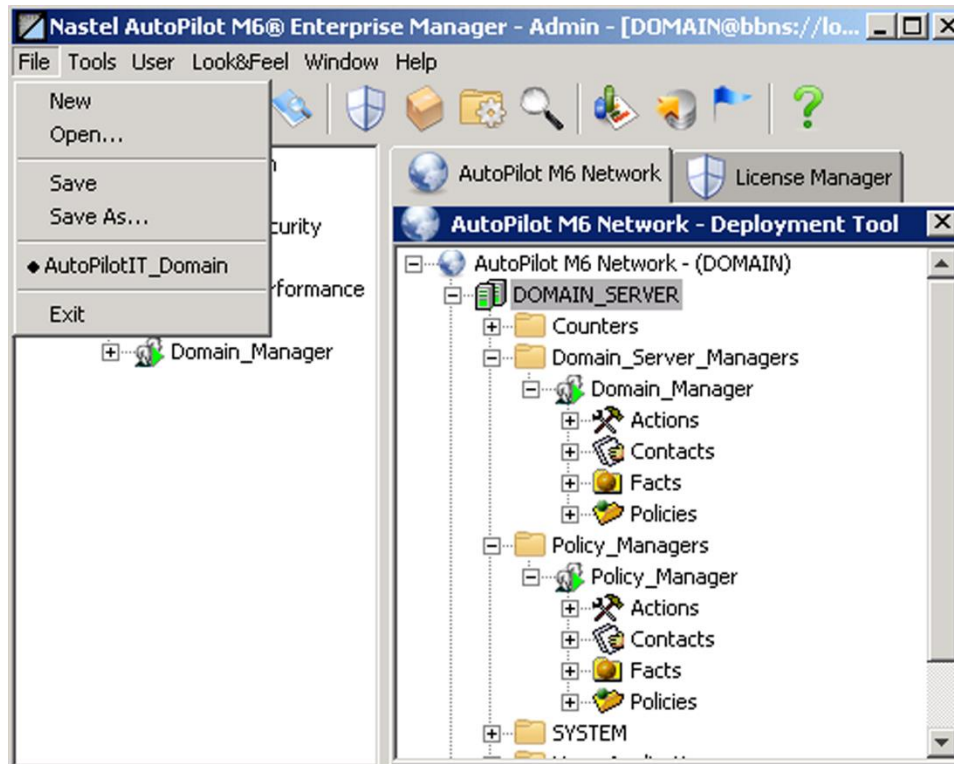


Figure 3-15. User Console - File Menu

2. **Tools:** Click **Tools**, *Tool Menu* is displayed. Clicking on an item listed in the Tools Menu will open the corresponding screen/view. Tool menu contents are same as the toolbar, with exception of the help screen, which is included on the toolbar. Click a tool to bring

it into view. Each screen has all the required links, menus, and toggles. As each screen is displayed, it is also listed in the Windows menu.

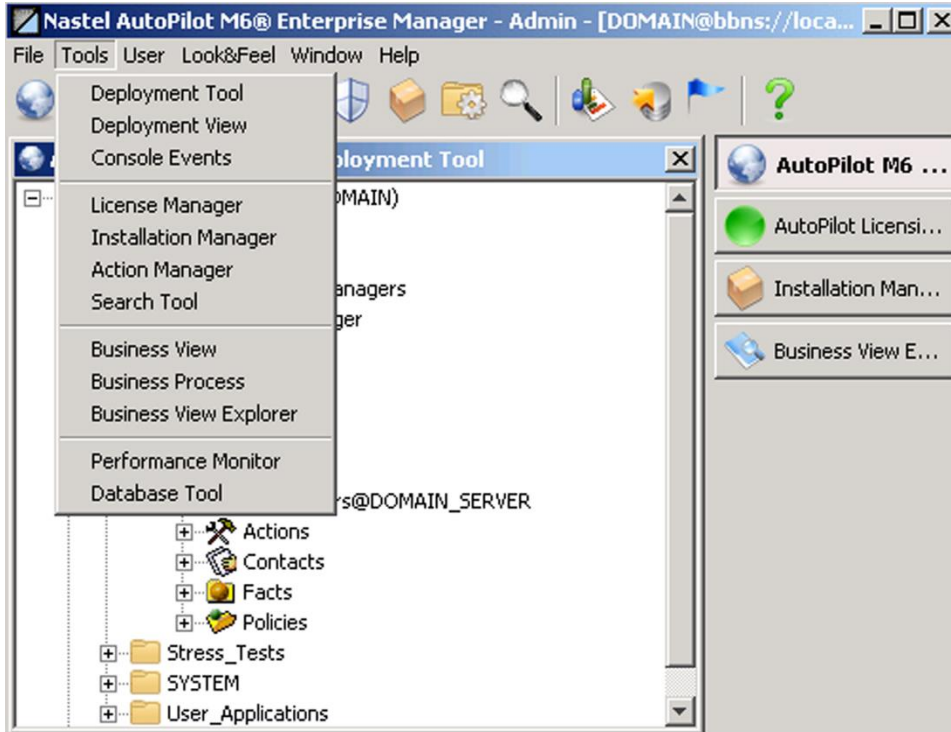


Figure 3-16. User Console - Tools Menu

3. **User:** Click **User** to display the pull-down menu.

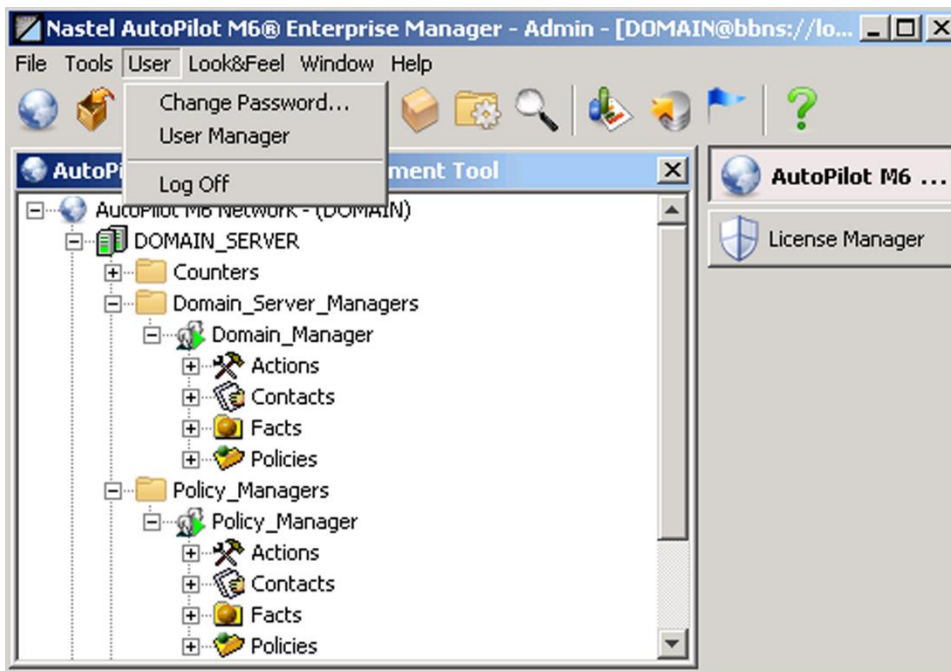


Figure 3-17. User Console - User Menu

- Click **Change Password** to display *Changing Password* dialog box. A user can only change passwords if your admin enables password security permission for that user.



Figure 3-18. Changing Passwords

- Click **User Manager** to display the *User Manager* screen. The **Users** list is displayed by default. Click **Groups** tab to display the list of groups.

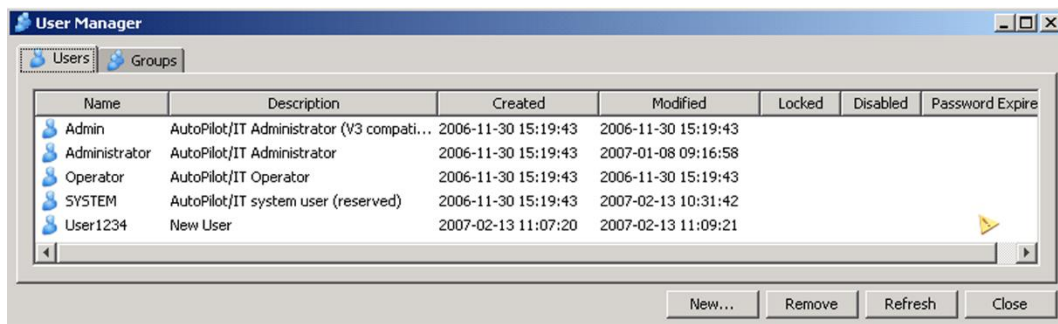


Figure 3-19. User Manager

- Click the **Groups** tab to display the list of user groups. Double-click a group or right-click and choose **Properties** to display the details for that group. There are four tabs: *General*, *Member of*, *Permissions*, and *Misc*. The admin manages the settings in these screens.

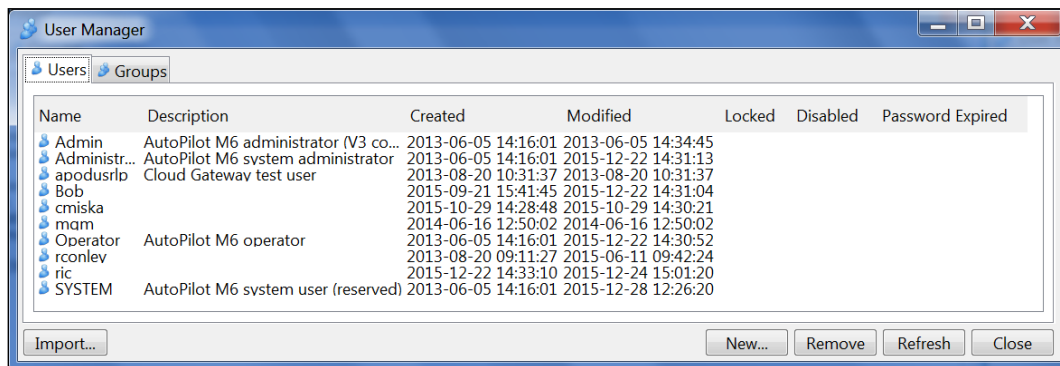



Figure 3-20. User Groups

 If the user belongs to an LDAP group, the Member of tab will show the group.

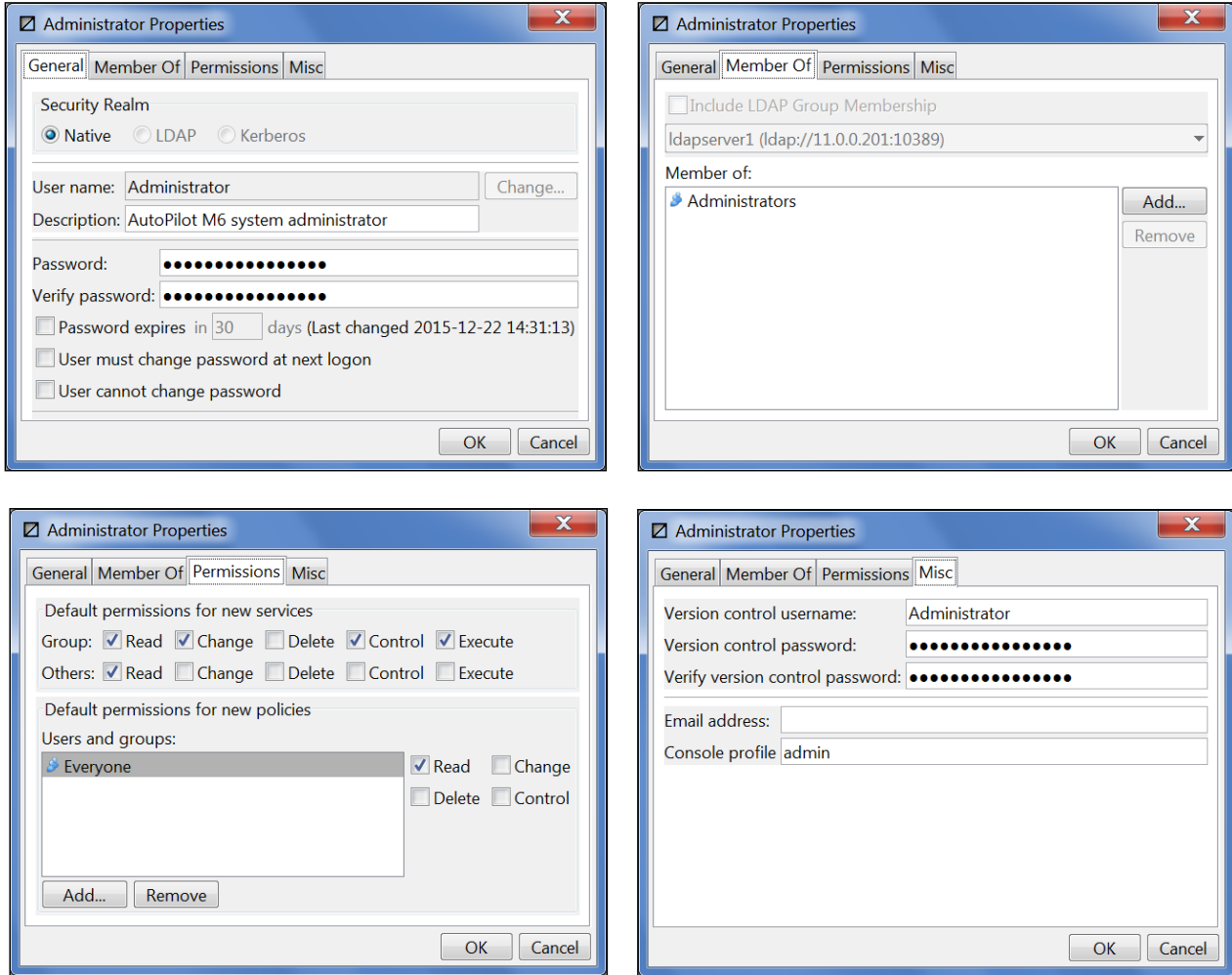


Figure 3-21. User Group Screen Details

4. **Look&Feel:** Click **Look&Feel** to display a pull-down menu of various ways to display the User Console.

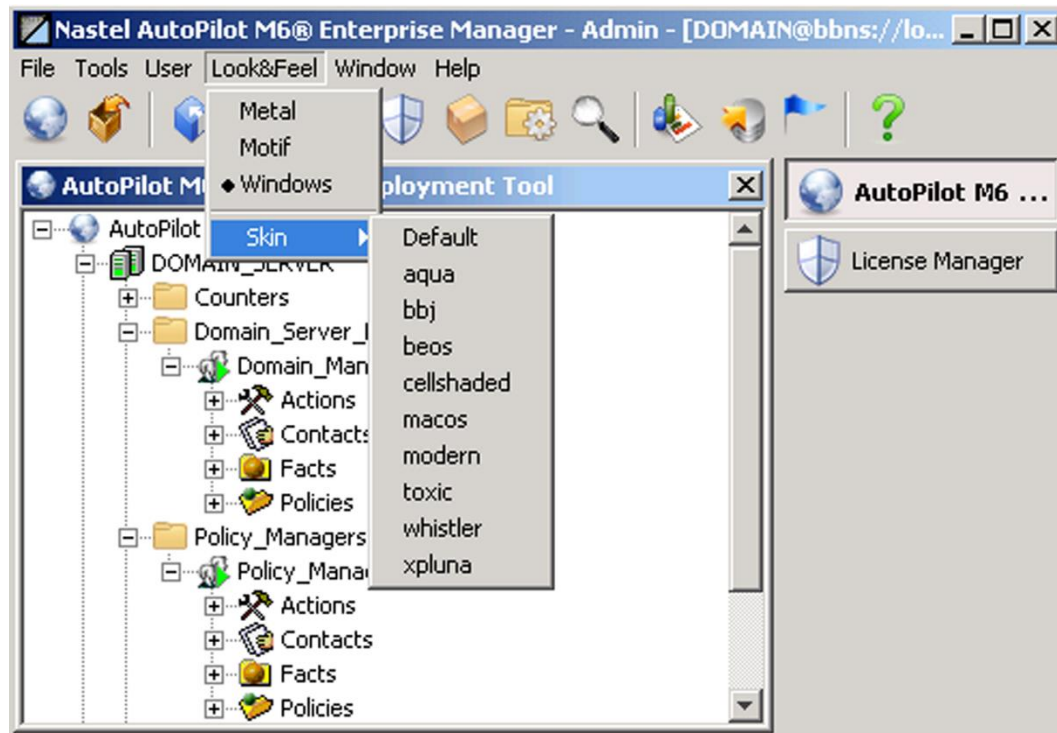


Figure 3-22. Look & Feel Menu

5. **Window:** Click **Window** to display a pull-down menu of open windows on the right side of screen. Clicking on a listed window will bring that window to the top of open screens.



Figure 3-23. Window Menu

- Click **Tabs** to toggle open window names between the top of screen and the right side of the screen.

- When open windows are separated from the console (drag and drop title bar to remove), the **Cascade** and **Tile** options are activated. Click the option you need to arrange the open windows as indicated.

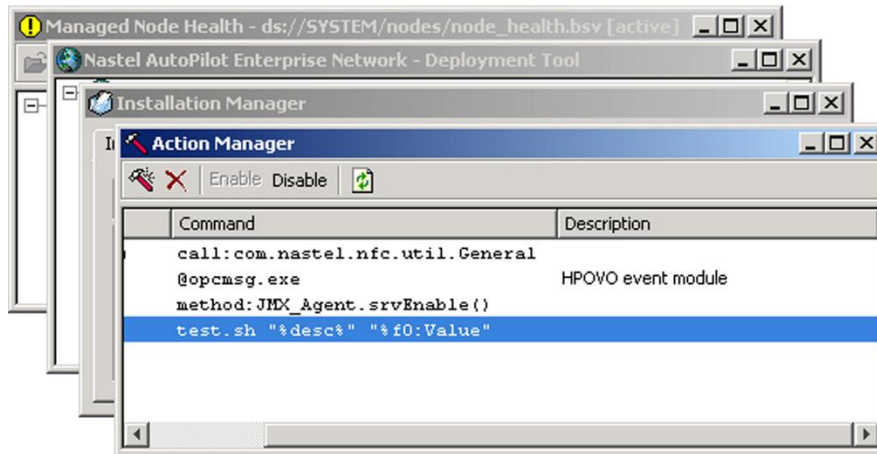


Figure 3-24. Cascading Open Windows

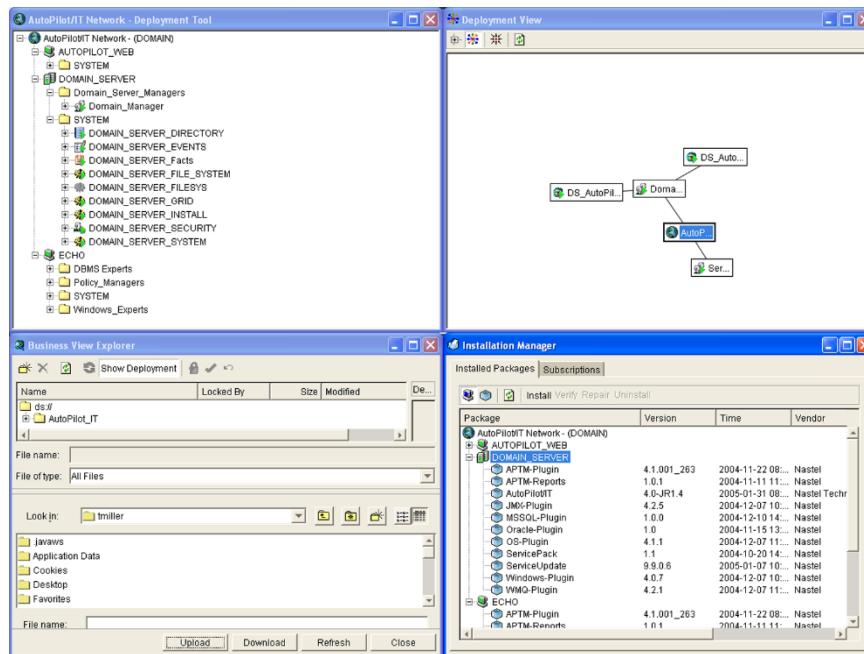


Figure 3-25. Tiled Open Windows

6. **Help:** Click **Help** to display pull-down menu of support topics.

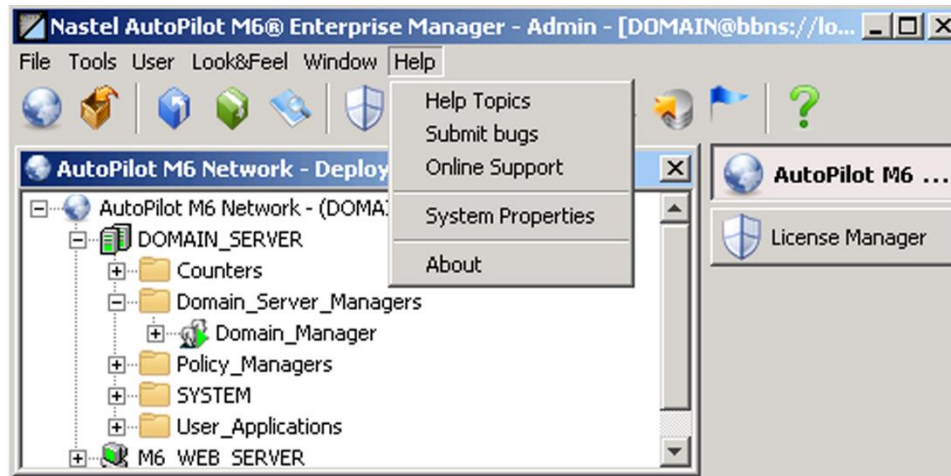


Figure 3-26. Help Menu

- Click **Help Topics** to access the meshIQ Resource Center.
- Click **Submit bugs** to open the meshIQ bug reporting screen. Username and Password are required.
- Click **Online Support** to access the meshIQ support system. Your *User ID* and *Password* login must be obtained directly from meshIQ. Please contact meshIQ support at mysupport@meshiq.com for more information. Check with your administrator about access privileges.

 A screenshot of the meshIQ mySupport Login page. At the top is the meshIQ logo, which consists of a green stylized owl head icon followed by the text "meshIQ™". Below the logo is the heading "Welcome". Underneath is the instruction "Log in to meshIQ to continue to mySupport." There are two input fields: "Email address*" and "Password*", with a toggle icon for the password field. Below the password field is a blue link "Forgot password?". At the bottom is a large green button labeled "Continue". Below the button is the text "Don't have an account? [Sign up](#)".

Figure 3-27. meshIQ mySupport Login

- Click **System Properties** to display the system properties screen. It contains all relevant Java system information. Scroll down to review all available system properties. This information will be useful for meshIQ support personnel in case of a problem.

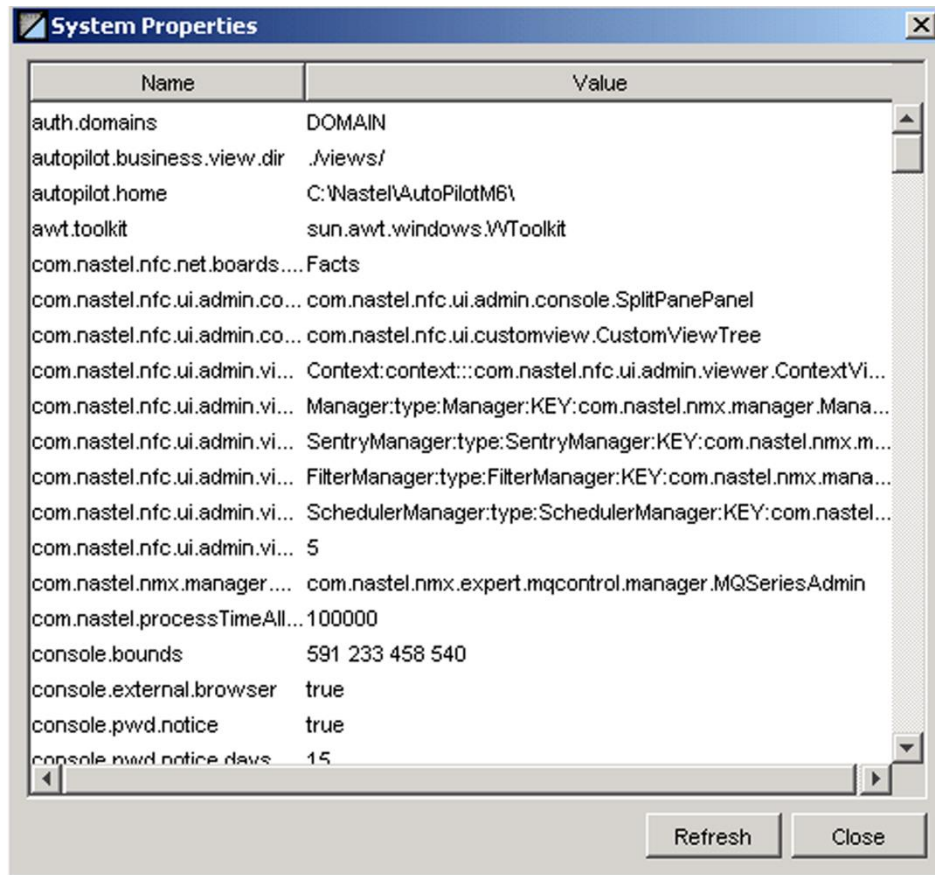


Figure 3-28. Console Java System Properties

- Click **About** to display version information. The information includes the version you have installed as well as the build number. This information is important when contacting meshIQ Support.

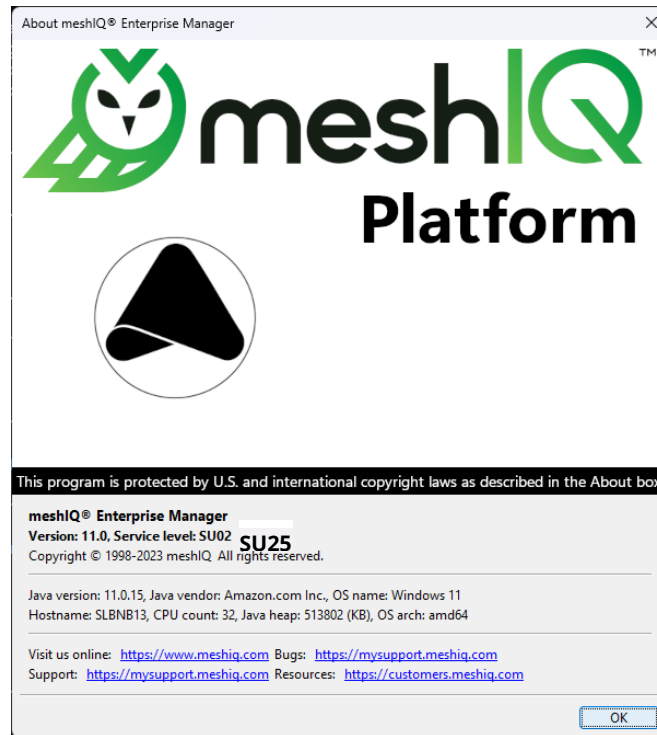


Figure 3-29. About meshIQ Enterprise Manager

Main Toolbar

The main toolbar icons correspond to the pull-down menu items.



Figure 3-30. Main Toolbar

Tab Selection

Easily navigate to previously opened screens by clicking on the tab for the screen you want to view. If you would like to view the tabs above the screen, as in earlier versions of Core Services, click **Window** to display the pull-down menu and then click **Tabs**.

3.2 Installation Manager

The Installation Manager, which is available through the User Console, allows users to manage software from one centralized point. A user can view, install, verify, and repair all packages from one screen. All packages are located under `ds://software` folder on the domain server.

Under the *Installed Packages* tab, there are two view screens:

- **View by Server** – what nodes within a domain the packages are installed on
- **View by package name** – name of package

Both screens display the Version Number, the Time the package was installed, the Vendor name, the File Name given to package, and the Size of the package.

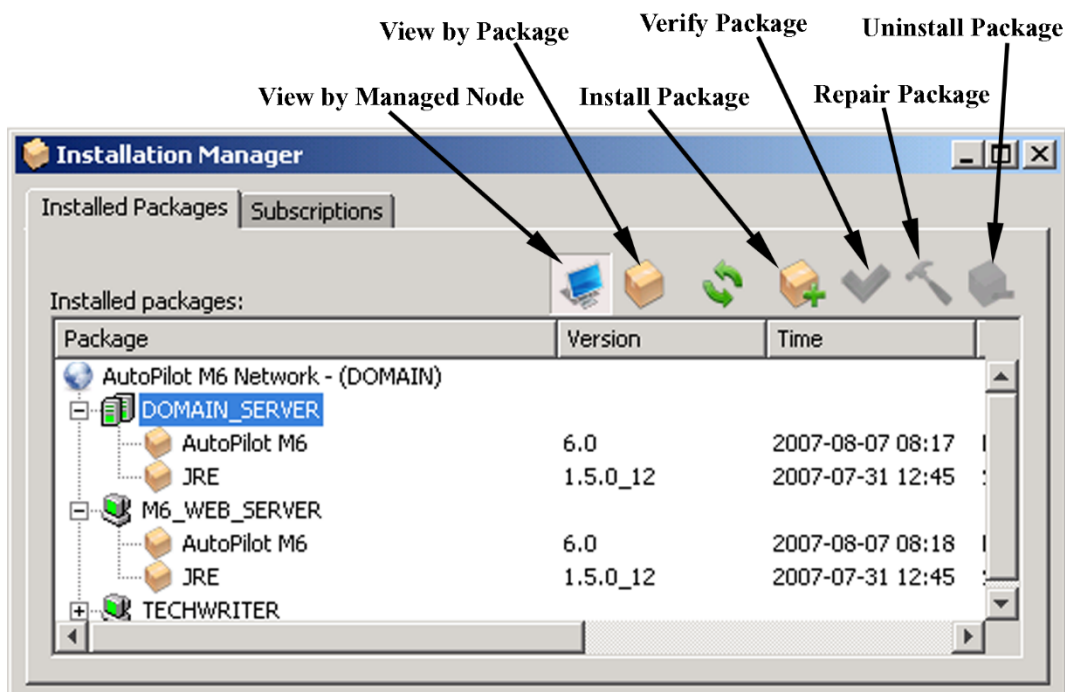


Figure 3-36. Installation Manager – CEP Server View

The *Subscriptions* tab allows the user to create update groups. These groups are a collection of updates that are automatically pushed out and installed on all active subscribers. Three update groups are provided. The user must uniquely configure each group by placing the appropriate files in each group.

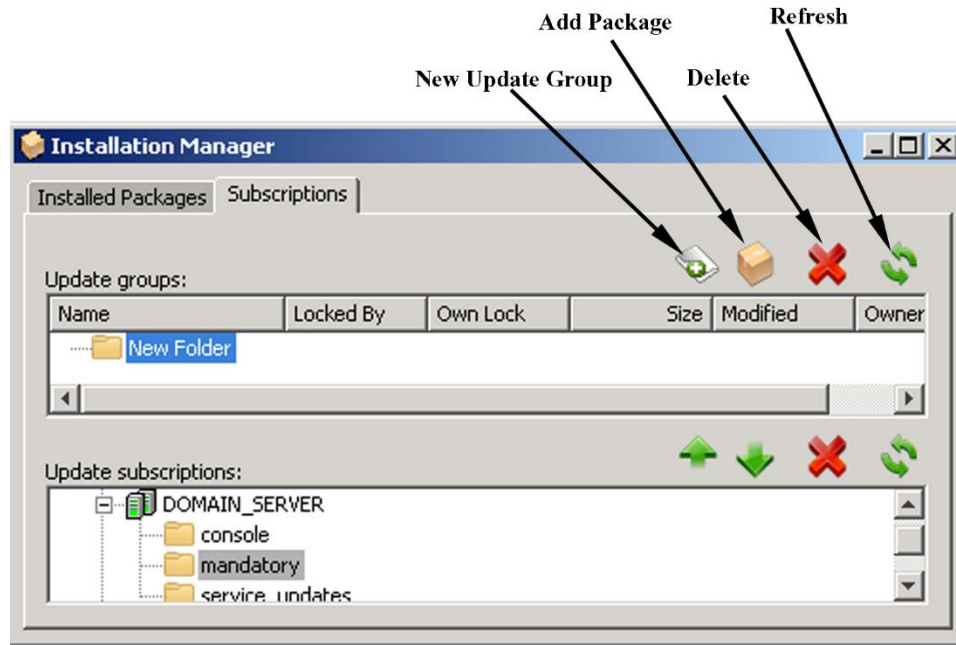


Figure 3-37. Installation Manager – Subscriptions Tab

Table 3-1. Update Groups	
Update Group	Description
console	Maintains updates designed for User Console installations only.
mandatory	Maintains updates that are mandatory for all meshIQ Platform Core Services installations.
service_updates	Maintains all Service Updates.

After startup, every user console, CEP server and domain automatically checks its list of subscriptions and downloads and installs all new and updated packages.

Chapter 4: Administering Core Services

This chapter describes administrative tasks and services. The steps described in this chapter are based on the default configuration; your configuration may vary based on installation, platform, and customization.



TIP

The "/" character is a reserved symbol and cannot be used as part of any name in meshIQ Platform Core Services.


4.1 Licensing

meshIQ has introduced a new licensing model for versions 11 and later that is based on the number of Total Instances and the User Count.

4.1.1 Obtaining Licenses

You will be contacted by meshIQ well ahead of your license expiration to begin the process of renewing your license. For information about meshIQ Platform licenses, see [About the meshIQ Platform license](#) in the Resource Center.

4.1.2 Checking the License Expiration Date

1. Click the license manager  icon from the main toolbar to display the *License Manager* screen.
2. Refer to the License Expiration Date field. It indicates the date when your license expires.

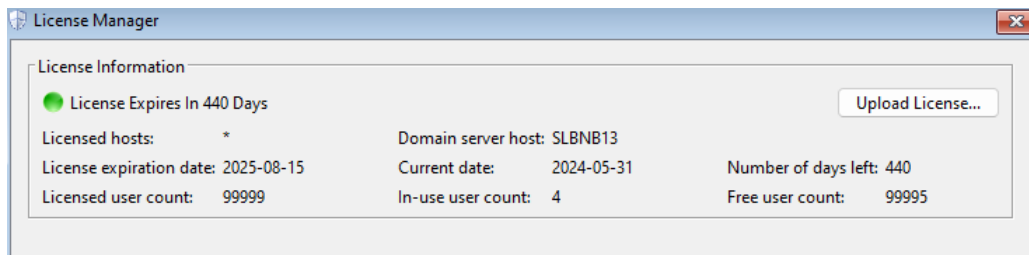


Figure 4-1. License Manager



Figure 4-2. License Expiration

4.1.3 Checking License Status

License status, limitations, and conditions can be reviewed using the *aplic* command. You should check the status after license updates are installed. From the command prompt type **aplic**. The current license information will be displayed.

```

CPU-License (../naming/meshiq_meshIQ_MQM.lic)
=====
Licensed Hosts : *
Licensed Users : 100
Expiration Date : 2025-08-15
Is Licensed Host : true, "PC3"
Is License Expired : false
    
```

Figure 4-3. APLIC Output

Table 4-1. License Status (APLIC)	
Property	Description
CPU-License	Location of license file.
Licensed Hosts	*
Licensed Users	Maximum number of users supported by the license
Expiration Date	current expiration date
Is Licensed Host :	true, "PC3" Indicates whether the host is licensed, followed by its name.
Is License Expired	whether the license is expired

4.2 Managing Users and Groups

4.2.1 User Manager

meshIQ Platform Core Services provides two built-in user groups and four built-in users. The *Admin* and *Administrator* (or those with Admin privileges) perform all management of the users and groups. The *SYSTEM* account is a reserved meshIQ account and used by system services and servers.

User Manager: Open the *User Manager* by opening the **User** menu, then selecting **User Manager**.

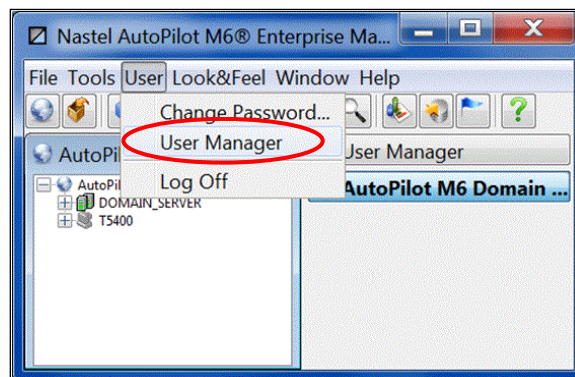


Figure 4-4. Opening User Manager



NOTE

When creating a new user or making changes to an existing user, the changes are logged to a file in the Domain Server. These audit records are viewed by right-clicking the Domain Server and selecting **View Events** from the popup menu.

The **User Manager** displays two folders: **Users** and **Groups**.

Users: The **Users** folder (Figure 4-5) is the default view. It lists all the registered users in your meshIQ network. It provides information about the user ([Table 4-3](#)) and links you to individual user profiles and status. Click a user record to view the user properties.

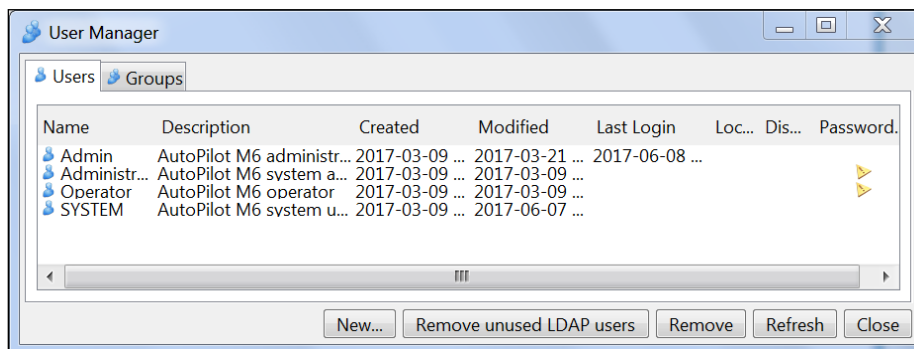





Figure 4-5. User Manager: Users Folder

Table 4-2. User Manager

Category	Description
Users folder	Displays a listing of all users registered in the meshIQ domain
Group folder	Displays a list of all user groups in the meshIQ domain
Import	Allows user or group definitions to be imported from an external security server (e.g., LDAP). Only available if LDAP support is enabled.
New	Allows you to add a new user or group
Remove unused LDAP users	Allows you to delete unused LDAP users
Remove	Removes a selected (highlighted) user or group
Refresh	Refreshes the User Manager
Close	Closes the User Manager

Table 4-3. User Manager: Users

Category	Description
Name	Default or Admin assigned user name.
Description	Descriptive information entered at the time the user was created.
Created	Time and date the user was created.
Modified	Time and date of the last modification to this user's profile.
Locked	The user is locked by the administrator or due to too many logon attempts. An exclamation point  icon under the locked column indicates a lockout.
Disabled	An exclamation point  icon indicates the user account has been disabled.
Password Expired	An exclamation point  icon indicates the user password has expired.

Groups: The *Group* folder (Figure 4-6) lists all users groups in your meshIQ network. There are two default user groups in the installation of meshIQ Platform Core Services: *Administrators* and *Operators*. Only the Admin (or those with Admin privileges) can add, delete, and modify the groups.

The *Group* folder provides information about the user group and links you to individual group profiles. The group list provides the following information listed in Table 4-4.

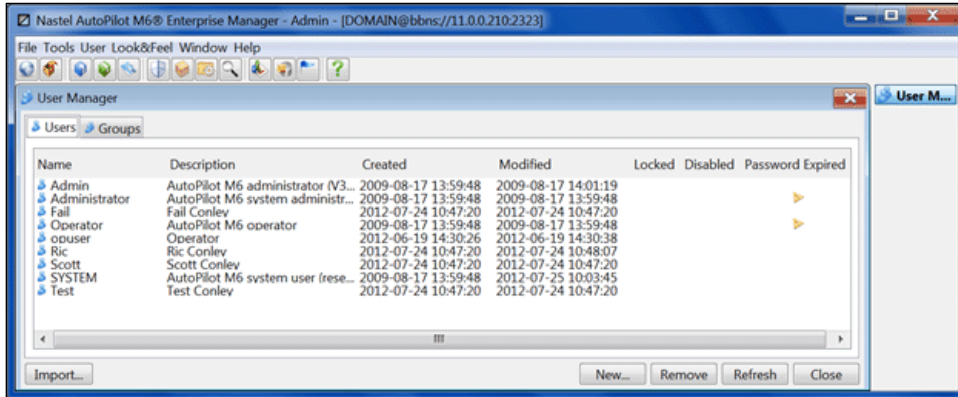


Figure 4-6. User Manager: Groups Folder

Table 4-4. User Manager: Groups	
Category	Description
Name	Default or Admin assigned user names.
Description	Descriptive information entered at the time the user was created.
Created	Time and date the user group was created.
Modified	Time and date of the last modification to this group's profile.
Locked	Not used for user groups.
Disabled	Not used for user groups.
Password Expired	Not used for user groups.

User Sub-menu

Right-click any user to open the sub-menu for that user.

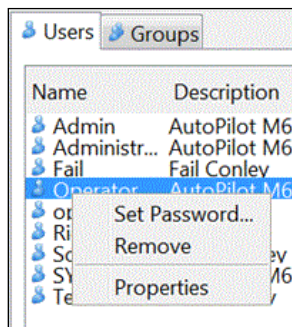


Figure 4-7. User Sub-Menu

Table 4-5. User Manager: User Options

Category	Description
Set Password	Opens <i>Set Password</i> screen (Figure 4-8). Enter user's new password in the <i>New Password</i> and <i>Confirm Password</i> dialog boxes. Password not available when opening group sub-menu. (Not enabled for LDAP.)
Remove	Deletes the selected user or group. The <i>Confirm Remove Account</i> screen is displayed, click Yes or No as needed.
Properties	Displays the selected user or group <i>Properties</i> screen (Figure 4-9).

Users with password update/change authority can change their own password. Restricted users must contact the meshIQ Platform Admin for password or username assistance.

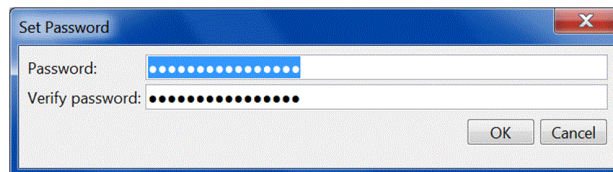


Figure 4-8. Set User Password

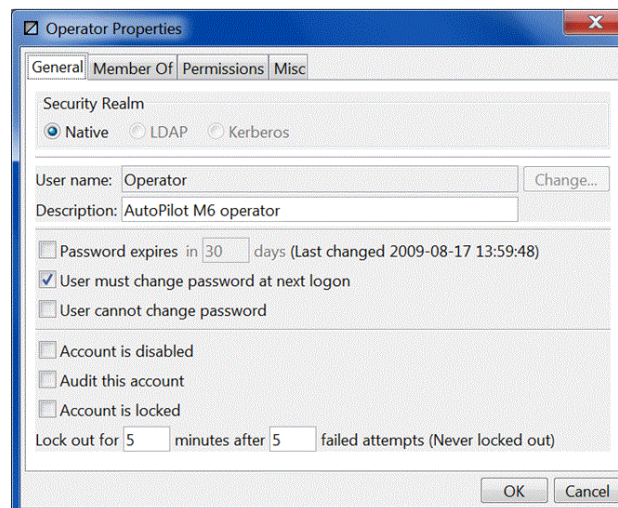


Figure 4-9. User Properties



TIP

The number of users that can be added and maintained is governed by your license. Consult your meshIQ Platform administrator or contact your meshIQ representative for further assistance.

The "/" character (forward slash) is a reserved symbol and cannot be used as part of any name in meshIQ Platform Core Services.

Group Sub-menu

Right-click any listed group to open the sub-menu (Figure 4-10). The sub-menu allows you to remove groups or access the group properties.

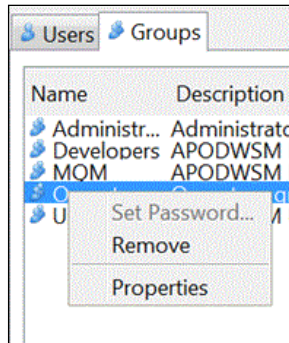


Figure 4-10. Groups Sub-menu

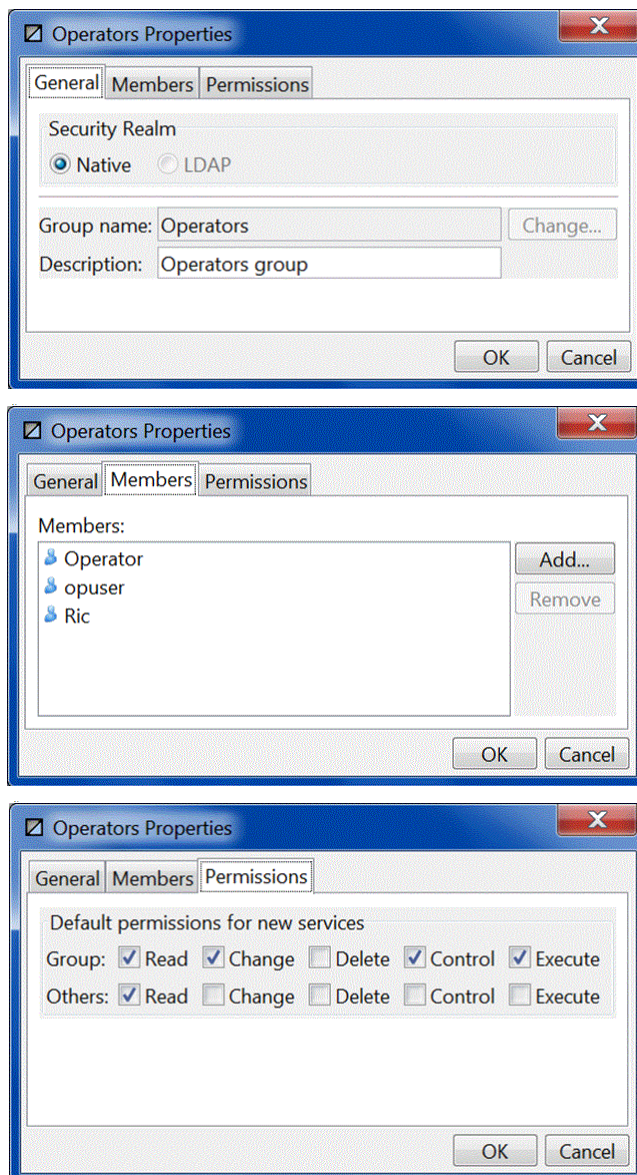


Figure 4-11. Group Properties

4.2.2 Adding Users

1. From the *User* folder ([Figure 4-5](#)) in the *User Manager* click **New**. The *New User* screen is displayed.

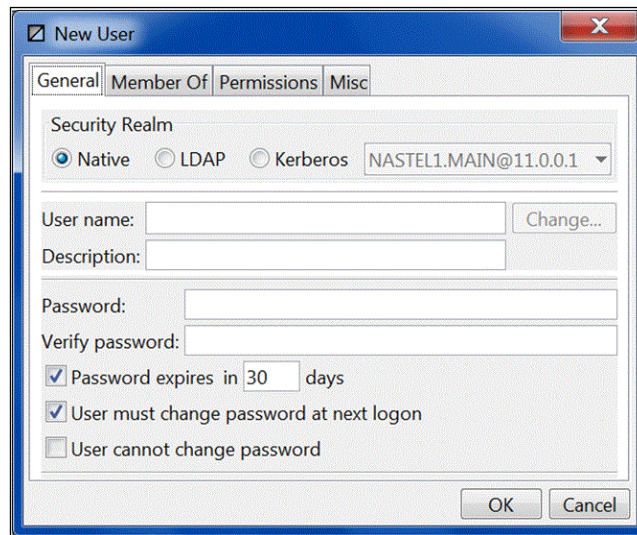


Figure 4-12. Adding New Users

2. Select either Native or LDAP.
 - Native:** proceed to step 3.
 - LDAP:** proceed to step 3.
3. **Native:** Enter the user properties as explained in the table in step 4 below.
 - LDAP:** Specify user name and description.
4. Fill in the following user properties:

Table 4-6. Adding New User: General	
Property	Description
Security Realm	Select the type of security model to be used when the new user is logging in.
User Name	User name in accordance with local policy.
Description	Logical user description based on local policy.
Change button	Click to display list of user names added when configuring LDAP.
Password/Verify password	Enter passwords. Password is Admin defined. Users can change as needed. See Changing Passwords in Chapter 4. (Not enabled for LDAP.)
User must change password at next logon	Click User must change password at next logon to enable/disable user password change at next logon. For Administrator Use Only. (Not enabled for LDAP.)
User cannot change password	Click User cannot change password to enable/disable user changing of password. For Administrator Use Only. (Not enabled for LDAP.)
Password expires in	Default is 30 days. Admin can define per local security policy. (Not enabled for LDAP.)

Table 4-6. Adding New User: General	
Property	Description
Account is disabled	Click Account is disabled to enable/disable user privileges. The check signifies that the account is disabled.
Audit this account	Click Audit this account to enable/disable user audit functions. The Audit function records all actions by the user on the server in the log file. The check signifies that the account audit is enabled.
Account is locked	Click Account is locked to enable/disable the user account. The check signifies that the account is locked.
Lock out user for	Locks out the user for an <i>Admin defined</i> number of minutes after a defined number of attempts to login have failed.

5. Click **OK**. Right-click newly created user and select *Properties*. Select the **Member Of** tab and click the **Add** button. The *Add User to Groups* screen is displayed.

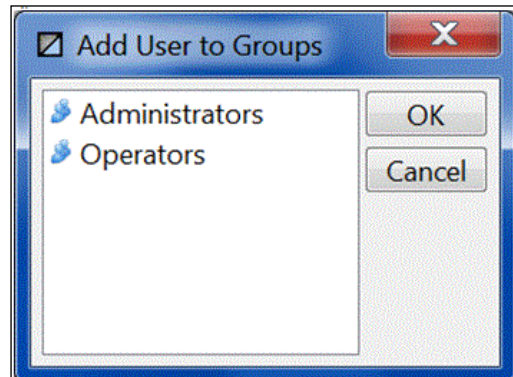


Figure 4-13. Assigning Users to Groups

6. Click the groups the new user is being assigned to. Click **OK**. Users can be assigned to multiple groups as needed. To select multiple groups, hold the **Ctrl** key and click the groups desired.

Table 4-7. Adding New User: Member of (Group)	
Group	Description
Administrators	System group that has administrative privileges
Operators	Group for the users' privileges
Others	Others are groups and users that do not belong to the accounts list of groups.

To have the user's group membership read from the LDAP Server, check the **Include LDAP Group Membership** box.

7. Click **OK**. Right-click newly created user and select *Properties*. Select the **Permissions** tab.

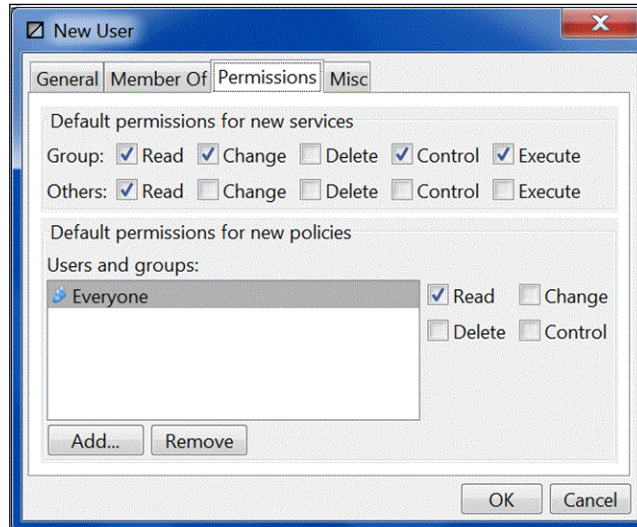


Figure 4-14. User Security Properties

8. Define the user permissions in accordance with local policy. Group permissions extend the checked privilege to all members of the specified user group. When the property is checked for others, users from other groups are extended the same privilege. All objects created by this user will automatically inherit its permission settings.

Table 4-8. Adding New User: Permissions		
Permission	Description	
	Group	Others
Read	Group members may read/view attributes of an object	Others may read/view attributes of an object
Change	Group members may change the attributes of an object	Others may change the attributes of an object
Delete	Group members may delete the object	Others may delete the object
Control	Group members may execute control actions such as start, stop, disable	Others may execute control actions such as start, stop, disable
Execute	Group members may execute operational commands on the object	Others may execute operational commands on the object

9. Click **OK**. Right-click newly created user and select *Properties*. Select the **Misc** tab.

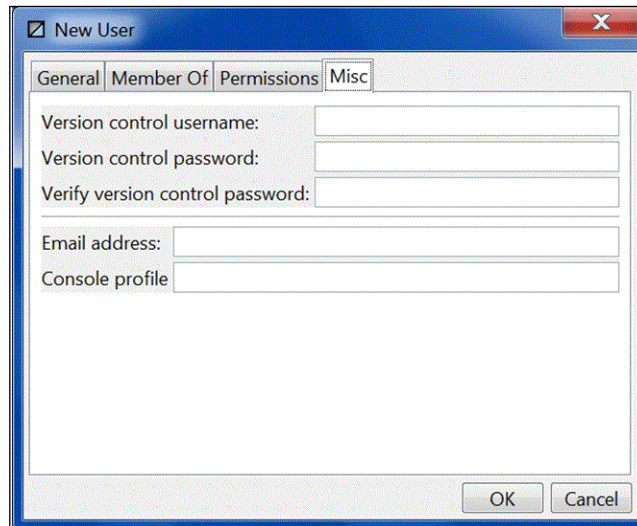


Figure 4-15. User's Business View Version Access

10. Define the new user's ability to update, add, or delete business views.

Table 4-9. Adding New User: Misc (Business Views)

Permission	Description
Version control username	Username assigned to those users requiring access to, and tracking of, business view history and revision.
Version control password	User password to authorize access and to add, delete, or change business views.
Verify version control password	Retype version control password.
Email address	User's email address.
Console profile	Specifies the name of the profile that the console loads based on user- defined profiles configured for each user to allow the console to be centrally managed. Maps to Domain Server profile directory which is stored under <code>\naming\profiles\<prof_name>< code="">. Profile must match <code>\naming\profiles\<prof_name>< code=""> directory. Each profile has <code>profile.properties</code> file which determines how the console is configured. A default profile is loaded for users whose console profile is empty.</prof_name><></code></prof_name><></code>

4.2.3 Adding Groups

1. From the *Groups* folder in the *User Manager* click **New**.

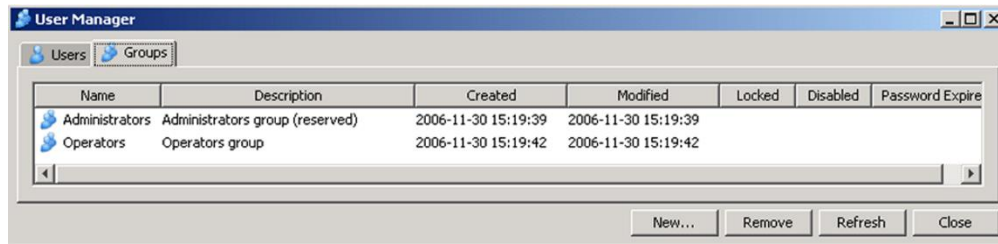


Figure 4-16. Adding New Groups

The *New Group* screen is displayed.

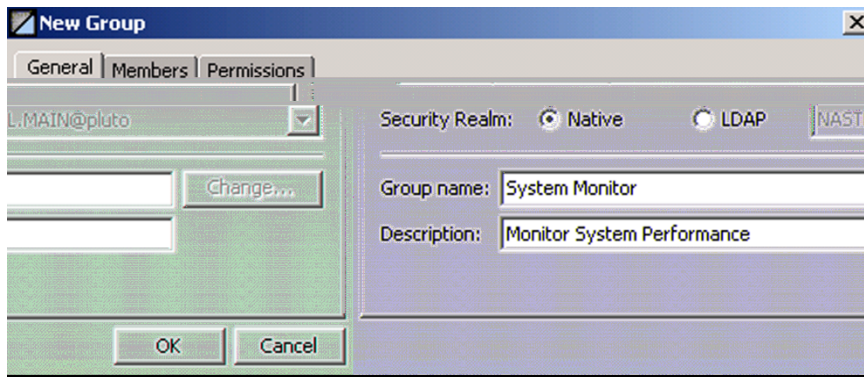


Figure 4-17. New Group General

2. Fill in the group name and description:

Table 4-10. Adding New Group: General	
Property	Description
Group Name	Group name in accordance with local policy
Description	Logical user description based on local policy.

3. Select the **Members** tab on the *New Group* screen.
4. Click **Add**. The *Add Group Members* screen will be displayed. Users can be assigned to multiple groups as needed. To select multiple members, hold **Ctrl** and click the groups desired. Click **OK** when all members have been added.

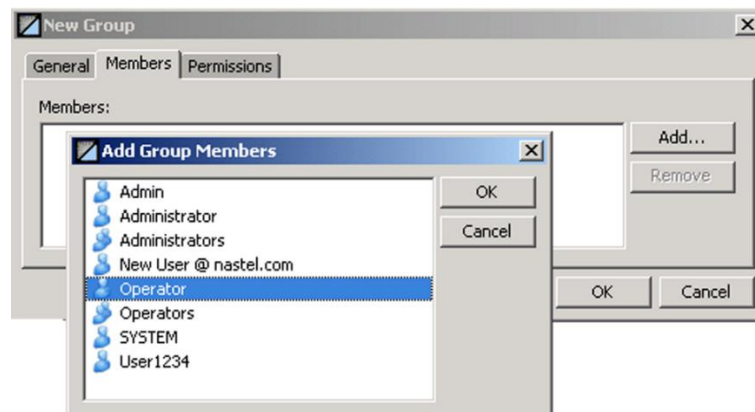


Figure 4-18. Assigning Members to New Group

- Define the user permissions in accordance with local policy. Group permissions extend the checked privilege to all members of the specified user group. When the property is checked for others, users from other groups are extended the same privilege. All objects created by this user will automatically inherit its permission settings.

Table 4-11. Adding Group: Permissions

Permission	Description	
	Group permissions	Others permissions
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, disable.	Others may execute control actions such as start, stop, disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.

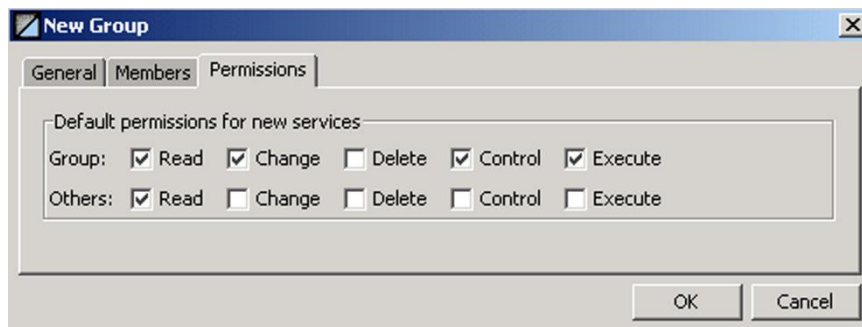


Figure 4-19. User Security Properties

4.2.4 Removing Users

The meshIQ Platform administrator can remove users as follows:

1. In the *User Manager*, *User* folder, click the user to be removed. Click **Remove**.
2. In the *Confirm Remove Account* screen click **Yes**. The user account will be permanently removed.

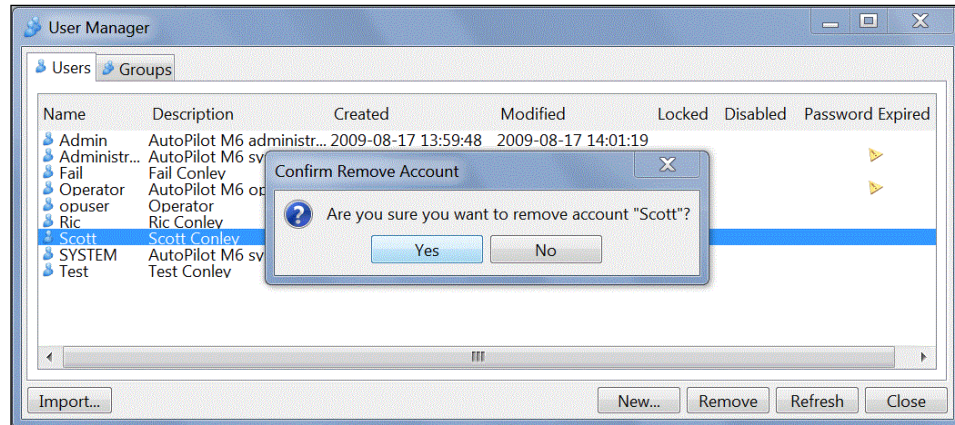


Figure 4-20. Removing Users

4.2.5 Removing Groups

The meshIQ Platform administrator can remove users as follows.

1. In the *User Manager*, *Groups* folder, click the group to be removed. Click **Remove**.
2. In the *Confirm Remove Account* screen click **Yes**. The user group account will be permanently removed.

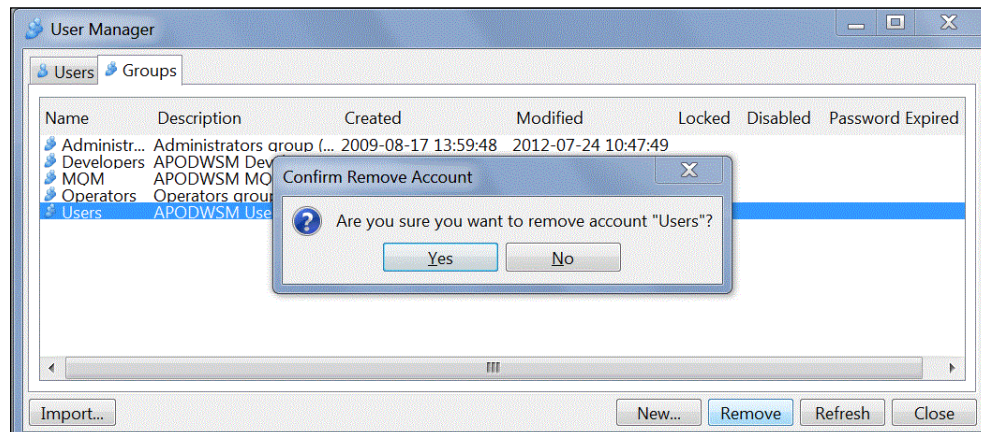


Figure 4-21. Removing Groups

4.2.6 Changing Passwords



This section applies to native users. It does not apply to LDAP users.

NOTE

Password and password format policies should be adopted and observed to prevent unauthorized system access. Default passwords should be changed as soon as possible to prevent unauthorized access. Both the user and the administrator can change user names and passwords. However, the admin can limit which users, if any, are permitted to change their passwords. The permission for users to access and change passwords is determined when the user is added and can be changed any time thereafter. Any password can be changed as needed or as dictated by local policy. Individual users can change their passwords as policy and existing permissions dictate.



Password length must be at least five characters long (by default) or as defined by `server.security.password.length=5` property configured at the domain server's `node.properties` file.

NOTE

There are several methods to change passwords based on your authority and permissions.

Changing Your Password

1. From the *User* menu, select *Change Password*. The *Change Password For <user>* screen is displayed.
2. Enter your current password in the *Current password* field.
3. Enter your new password in both the *New password* and *Verify new password* fields.
4. Click **OK**.

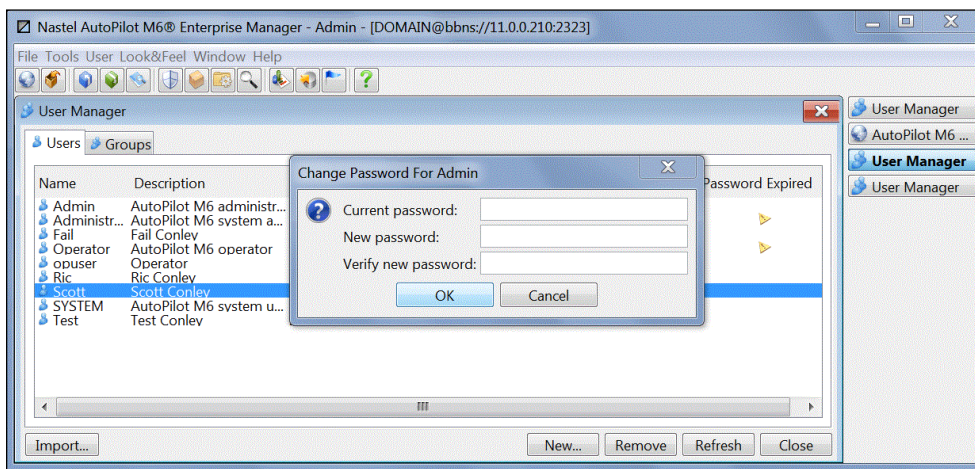


Figure 4-22. Changing User Passwords

Changing User Passwords

1. From the *User* menu select *User Manager*.

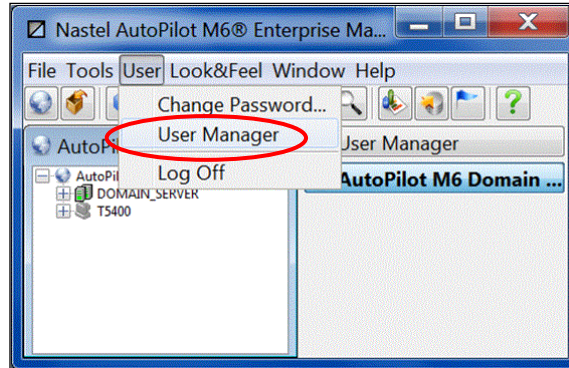


Figure 4-23. Open User Manager

2. From the *User Manager* menu, right-click the target user. Click **Set Password** on the sub-menu. The *Set Password* screen is displayed.

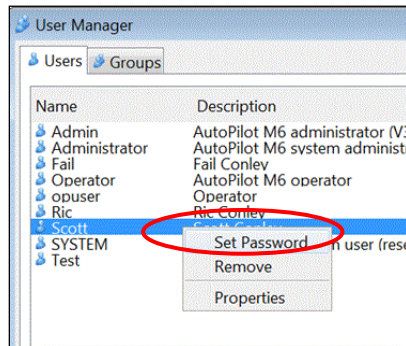


Figure 4-24. Set User Password

3. Type new user password in the *password* and *Verify password* boxes. Click **OK**.

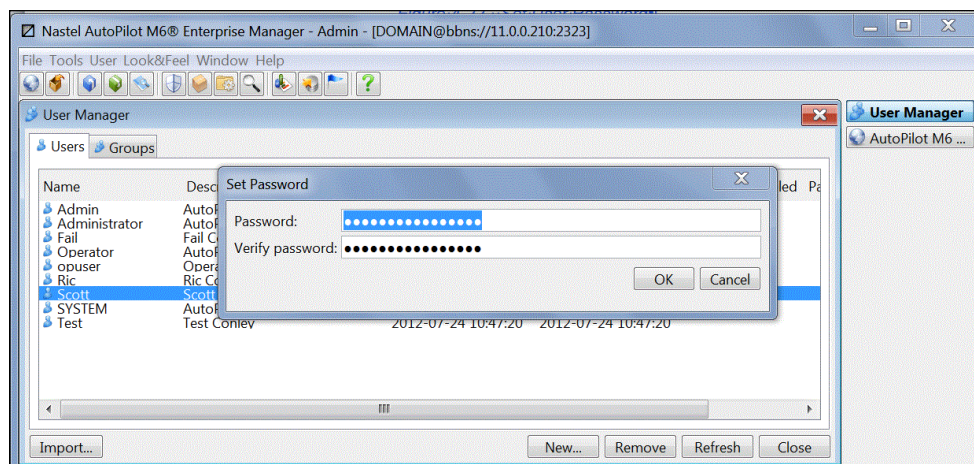


Figure 4-25. Change User Passwords

4.2.7 Importing Users and Groups

If LDAP support is enabled, an **Import** button is available which will allow user or group (based on current view) definitions to be imported from the configured LDAP servers.

To import users and/or groups, click the **Import** button to open the *Import Users or Import Groups* screen.

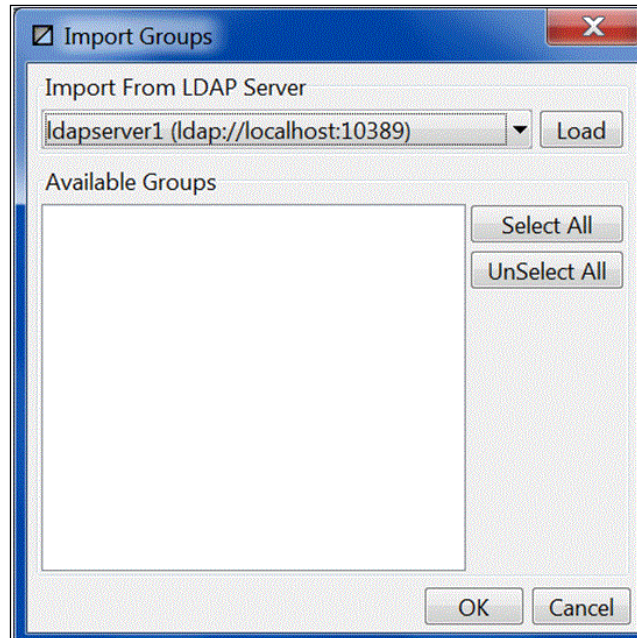


Figure 4-26. Import Users or Import Groups

1. At the top, pick from the configured LDAP servers and click **Load**.
2. The **Available Groups** list will then be populated with all users or groups found at the selected server whose names do not match those of existing users or groups.
3. Select the users or groups to import and click **OK**.

4.3 Service and Account Permissions

4.3.1 Service Permission Mask

The security settings are common for all management services. However, the actual settings may vary depending on the owner's permission mask. By default, all services inherit permissions from the owner account.

Table 4-12. Service Permission Settings	
Inherit permissions from owner	Clicking the check box enables or disables the transfer of permissions from the original owner to the owned object. The default is enabled (checked).
Owner	Name of the account that owns the object. The owner can be changed by clicking Change .
Permissions	The default mask is defined in the group and other permission settings in your user properties. (Refer to Adding Users .) The owner or anyone who has change permission can change security settings.

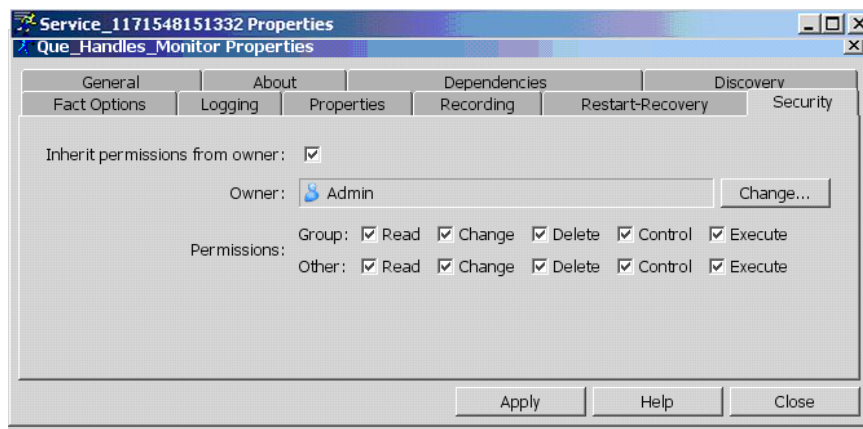


Figure 4-27. Service Permission Mask

4.3.2 Account Permission Mask

Permissions specify access control to objects by group members and those who are not part of the group (others). Owners always have full access to all owned objects. Permissions are inherited from the owner to the owned service such as expert, manager, and policy.

1. The default settings are the same for all new users and groups. The meshIQ Platform administrator must define the security settings for each group, and if needed, each account.
2. Permissions in the Group and Other categories are enabled or disabled by clicking on the appropriate button. The check mark indicates that the permission is enabled.

Table 4-13. Group and User Permissions		
Permission	Description	
	Group	Other
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, disable.	Others may execute control actions such as start, stop, disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.

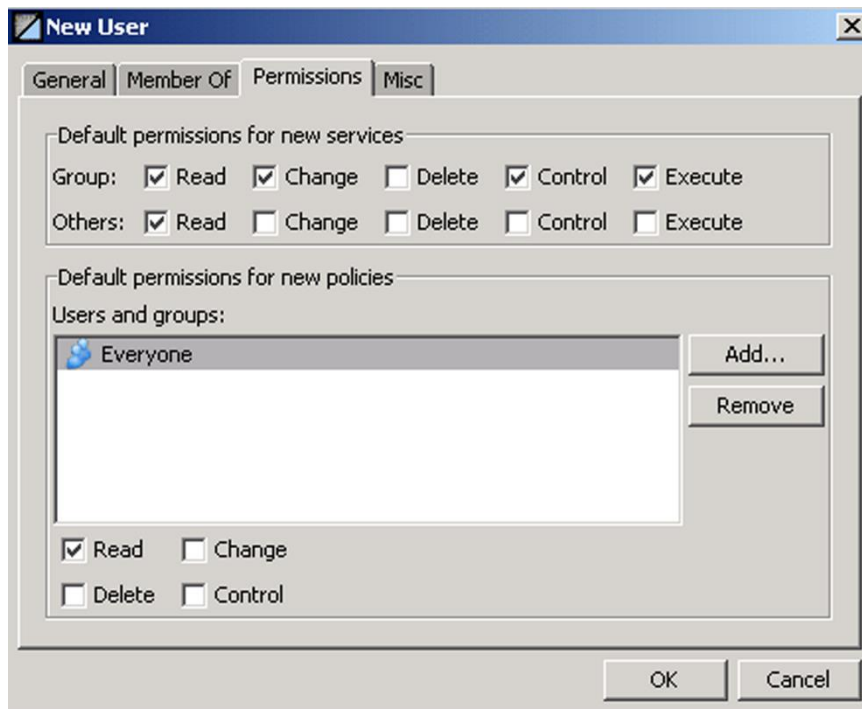


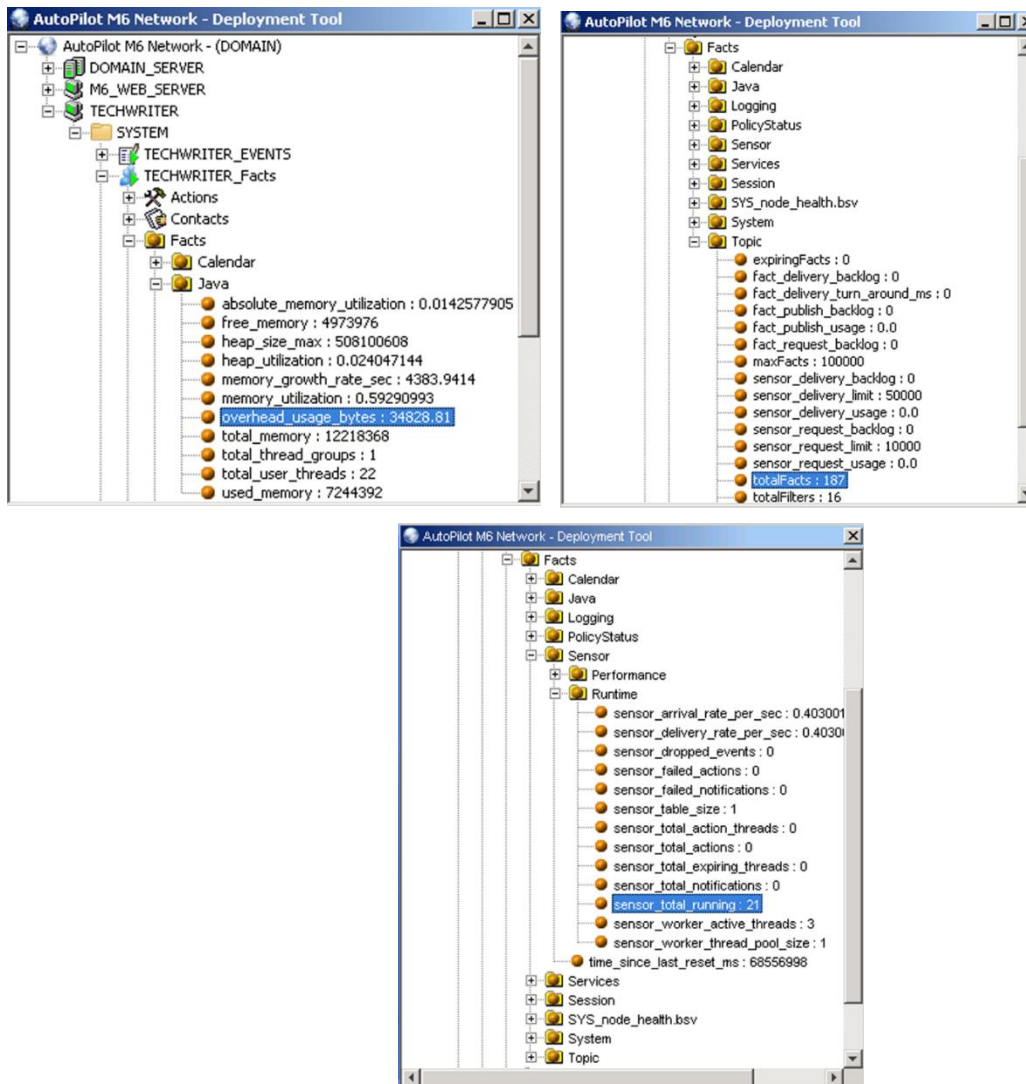
Figure 4-28. Default User Permission Mask

4.4 CEP Servers

A CEP server is a meshIQ Platform target-managed resource that holds services such as experts, managers, policies, and sensors. There are also built-in CEP servers that are available immediately after installation such as the Domain Server. From a CEP server you can view events, import services, deploy experts and managers, and refresh security (permission settings) to corresponding services. The CEP server icon reflects the status of node_health.bsv to help users spot CEP servers that have performance or resource issues. The icon will remain green unless the CEP server falls into a warning (yellow) or higher state. Stopping a CEP server stops all services associated with it.

4.4.1 Performance Guidelines

Every CEP server publishes information about its memory and fact utilization. This information is published under [NODE_NAME]_Facts\java and [NODE_NAME]_Facts\Topic. These facts can be included in business views and can be monitored and alerted on.



Memory Utilization and Estimation

For each type of environment, the total memory utilization in bytes depends on what is required for objects and facts. The amount of memory required will vary depending on the size and number of objects in the environment and the number of facts per object. meshIQ's CEP server configuration tiers take into account JVM Heap Memory slack. meshIQ provides a calculator to assist you with estimating memory utilization for your environment.

4.4.2 CEP Server Communication

The `[server]_Facts` monitor publishes new metrics about CEP server communications as follows:

Table 4-14. CEP Server Communication Metrics

Metric	Description
Session\total_bytes_sent	Total bytes sent across all pipes.
Session\total_objects_recvd	Total received objects across all pipes.
Session\total_objects_sent	Total sent objects across all pipes.
Session\total_objects_dropped	Total dropped objects due to one or more of the pipes being disabled.
Session\total_pipes	Total number of allocated pipes.
Session\total_pipes_disabled	Total number of pipes that no longer accept outbound traffic.
Session\total_pipes_full	Total pipes are filled to 100% capacity (max_send_queue_limit).
Session\total_pipes_idle	Total pipes that are currently idle.
Session\total_send_queue_depth	Total number of objects queued for outbound communication across all pipes.
Session\max_send_queue_limit	Maximum capacity of each pipe.
Session\avg_pipe_usage	Average usage for all pipes.
Session\avg_pipe_arrival_rate_per_sec	Average number of objects being queued for outbound communication.
Session\avg_pipe_delivery_rate_per_sec	Average number of objects being sent out per second.



IMPORTANT!

It is important to monitor `Session\total_send_queue_depth` and `Session\total_pipes_disabled`.

- `Session\total_send_queue_depth` is an indication of slow connections that cannot keep up with the rate at which facts are generated.
- A pipe is disabled once it reaches 100% capacity utilization. Once a pipe is disabled, it can no longer accept any new traffic and begins to drop any new activity (`total_objects_dropped`).

4.4.3 Using CEP Servers

Right-clicking on CEP server displays the following sub menu. CEP server options are described below.

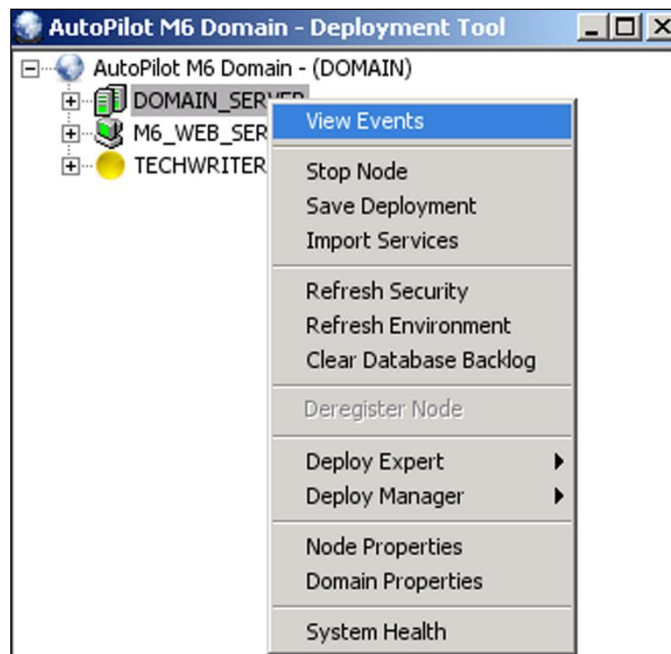


Figure 4-29. CEP Server Options

Table 4-15. CEP Server Options	
Option	Description
View Events	Displays the Event Viewer and all events associated with the selected node.
Stop Node	Turns off the CEP server and all of its services.
Save Deployment	Saves the configuration of the CEP server selected.
Import Services	Processes import files. Details are contained in the Event Log.
Refresh Security	Refreshes security (permission settings) to corresponding services. The settings are applied from the owner's permission mask. This option removes the need to restart the CEP server when account security mask is changed.
Refresh Environment	Reloads all previously loaded property files used by business views (for example, global.properties and node.properties).
Clear Database Backlog	Clears database backlog to free up memory. This option should only be used when <i><managed_node>_Facts\Topic\sensor_dbpool_backlog</i> fact is growing too quickly due to persistent database failures. Backlog can also be cleared from the APNET command line as follows: <pre>apnet invoke [managed_node_name]_SYSTEM.clearDBBackLog()</pre>
Deregister Node	Removes node from database. Only enabled after node has been stopped.

Table 4-15. CEP Server Options

Option	Description
Deploy Expert	Makes Expert (Samples, Probes, or Wrappers) ready for use.
Deploy Manager	Makes default policy manager ready for use.
Node Properties	Displays node property configurations.
Domain Properties	Displays domain property configurations.
System Health	Displays node_health.bsv which monitors CEP server health and performance.

4.4.3.1 Grid Support

The meshIQ Platform provides grid support for all CEP servers. The grid is a collection of highly available services that allow the user to provide uninterrupted management in the event of server failures. To implement a grid, do the following:

1. On the server running the Domain Server, create a folder named grid under `$AUTOPILOT_HOME/naming`. (`$AUTOPILOT_HOME` is the install directory of meshIQ Platform Core Services.)
2. Create a folder for the grid you are defining under `naming/grid`. (The folder name is user-defined and should accurately represent the collection of services you will be placing in the grid.)
3. Create all the required services on a dummy CEP server.
4. Copy `registry.xml` to the `naming/grid/[grid_name]`.
5. Create a file named `server.properties` and enter all the required properties to be used by the business views. (For example, required properties could be environment variables.)
 - `registry.xml` – defines shared cluster services
 - `server.properties` – defines properties required by cluster services



To be selected as the primary server by the grid manager, each CEP server must declare a priority. The higher the priority, the more likely it will be selected as the primary server.

6. Start all participating nodes using the following parameters:


```
ATPNODE -console -join [grid_name] -i [number]
```

(`<number>` is the priority.)

You may also provide these parameters through the CEP server's `node.properties` file. For example, for CEP servers running as Window Services:

```
property server.grid=[grid_name]
property server.grid_enabled=true
property server.grid.pri=0
```

The Grid Manager is always running within the domain server, and it must be highly available. When the domain server is started, the grid manager waits for CEP server registrations with a specific grid and then begins the primary election logic based on CEP server priority.

Each grid has only one primary server. Grid manager votes on the primary server based on the CEP server priority when the primary server becomes unavailable for any reason. When the primary server becomes available and joins the grid, it will not become primary until the current primary server is taken offline. The following properties control primary voting logic:

Grace period in ms before electing primary – gives a primary server a chance to rejoin and regain primary status.

server.grid.vote.delay where default is 1500 – delay vote in ms.

Each grid must have its own primary server and should be checked periodically by using:

server.grid.vote.sample where default = 120000.

4.5 Experts

In meshIQ Platform Core Services, experts are agents that are knowledgeable about the subjects they are assigned to monitor. Experts are mobile and act as connectors to other experts. They are contained in target managed resources (nodes) and collect system status information, which is analyzed by assigned policies and business views: they are filtered by managers to be viewed in a business view.

Experts publish facts. Facts are basic true statements about current run-time values, indicating an object's state in a managed resource.



- NOTE**
1. By default, users in the Operator Group have read-only and operational access to expert properties. If you require additional access privileges, contact your local meshIQ Platform Admin.
 2. The number and type of built-in experts will vary based on options exercised at installation.

Experts perform the following functions:

- Experts serve as liaisons between managed resources and the meshIQ Platform
- Understand events, information, and protocols of managed resources
- Translate unique “languages” of managed systems to meshIQ Platform Core Services facts
- Collect and publish facts about managed resources to all interested subscribers
- Carry out actions requested by users via the User Console.
- Carry out actions requested by managers or policies.

4.5.1 Built-in Experts

The Application Instrumentation Module (AIM) which is described in this section is a licensed function. When used for data collected by Core Services or by one of the licensed experts, this function is an extension of that license. Additional licenses are required when AIM is used to collect data from other products or components. Examples include using the SQL

expert to collect data from a database created by another product or using APFACT to send data to the process monitor. meshIQ Core Services includes the following built-in experts:

- Samples
- Probes
- RSS News Feed
- Wrappers



There are many additional built-in experts that are supplied with various plug-ins. See the guide for the appropriate plug-in for additional expert details.

4.5.1.1 Samples

The following sample experts are provided as references and can be used by developers to create their own experts.

- **Counter:** Increments an integer based on a user-defined timer.
- **Car:** Demonstrates a simple car simulation.
- The source code is available in `[AUTOPILOT_HOME]\examples` directory and is available when “*SDK Samples*” option is installed.

4.5.1.2 Probes

meshIQ Core Services has three built-in probes as follows:

- URL Monitor
- Stress Test
- SQL Query Monitor

URL Monitor

The URL Monitor measures the availability and performance of one or more URL addresses such as files, web pages, and web applications.

The URL Monitor accepts one or more URLs, such as <http://www.meshIQ.com> and <https://www.google.com> and can include parameters as appropriate (?id=9999). If required, a proxy can be configured to access the URL.

Based on the sampling rate, the URL Monitor will attempt to connect to each URL configured (as a synthetic transaction). If successful, it will return details about the response packet. If unsuccessful, it will return details about the results and the exception. (See figure below.)

The facts vary based on the response from the expert. Descriptions of the various response facts can be found at the following web pages:

- <https://www.rfc-editor.org/rfc/rfc9110.html>
- http://en.wikipedia.org/wiki/List_of_HTTP_header_fields

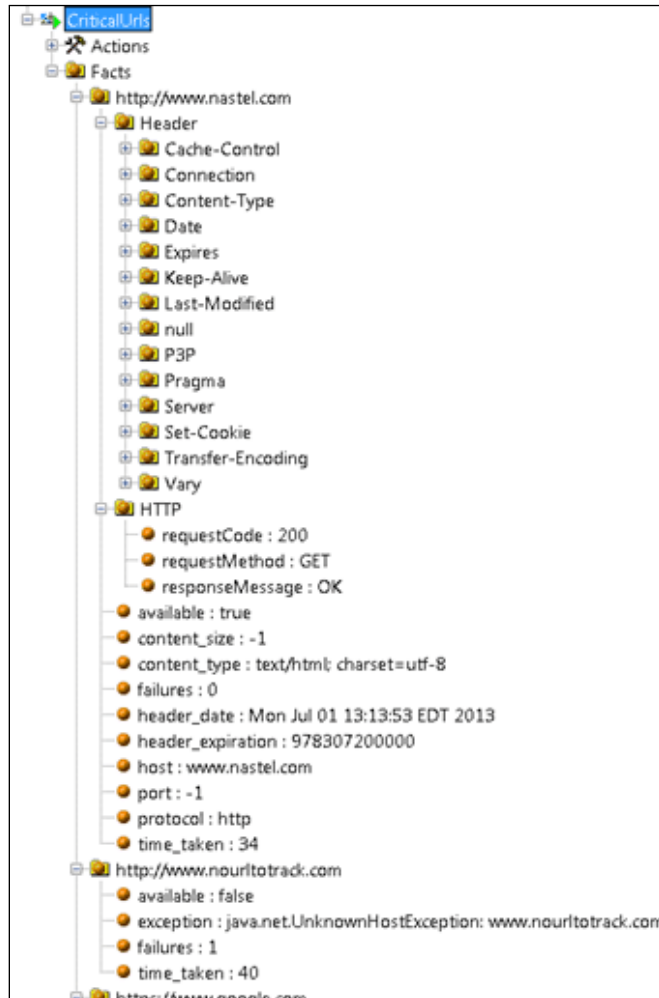


Figure 4-30. URL Monitor Success Data

The user can specify content string(s) that must be present in content by selecting **Load content** and specifying the content string(s) in the **Search Content** field in the figure below.

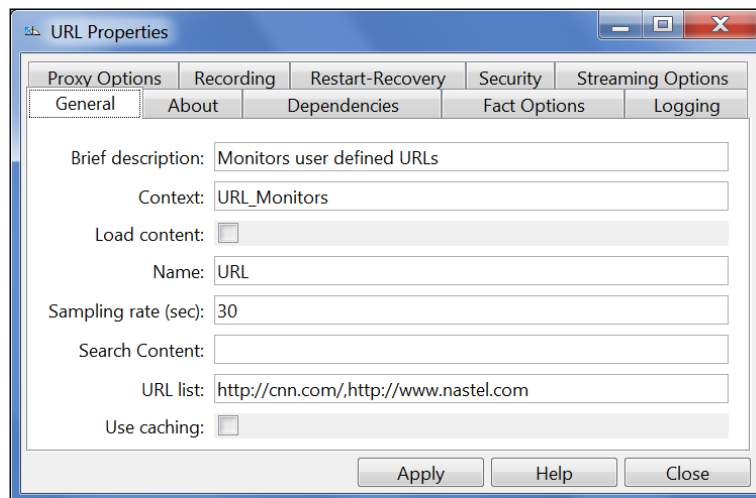


Figure 4-30A. URL Properties - General

Table 4-15A. URL Properties – General	
Property	Description
Brief description	A short, user-defined, description of the service
Context	A user-defined category that will be registered with the domain server. The default is: <i>URL_Monitors</i> . Context can be changed as needed.
Load content	Load URL content during the scan (used in conjunction with Search Content field).
Name	Name that uniquely identifies the service in the domain
Sampling rate (sec)	Sampling rate in seconds
Search Content	Comma-separated list of content to search for. Load content field must be selected. One or two facts are created (Figure 4-30B) : content_match – true/false content_line – data from url that matched (not present if content_match = false)
URL list	Comma-separated list of URLs to be scanned
Use caching	Document cache when sampling URLs (protocol dependent)

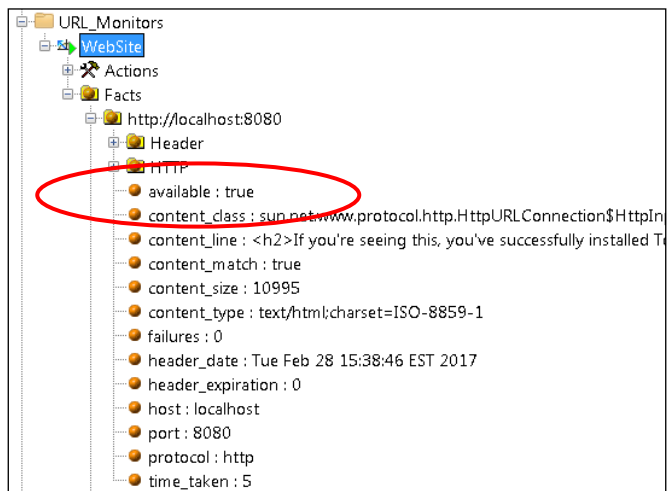


Figure 4-30B. New Facts Created

Stress Test

The stress test allows users to generate hundreds or thousands of facts to stress test CPU and memory performance of a CEP server. This probe can be used to test performance and scalability of a CEP server. Flow control is provided for all socket communications by preventing network delays from degrading CEP server CPU and memory performance using the following properties:

- **server.pipe.processor.limit=20000**. When the maximum number of queued objects for each socket connection reaches its limit, the connection is disabled, and objects are dropped.

- **server.pipe.delivery.flowpct=60.** Setting the value between 0 and 100 temporarily suspends the flow when the usage reaches 100%. The flow will remain suspended until the usage drops below the value specified by the variable.
- Setting the value to 0 or below disables this property, which means the flow will always stay enabled. Enabling flow control puts a maximum limit on the number of outstanding events for each connection and limits the memory growth associated with the unlimited growth of queued objects.

SQL Query Monitor

The SQL Query Monitor allows users to define an SQL query and publish the results to Core Services. By default, it is configured to sample MySQL STATISTICS table with a sample query.

When specified, the **Sort By Column(s)** field under *SQL Query* tab controls the instance creation. The SQL Query creates a set of facts for each row. The facts represent the row returned. For example: If select a,b,c from db.table where a >100, a set of facts is created for each row that matches a >100 and 3 facts (a, b, and c) if **Sort By Columns** was specified.

To create an SQL query, the expert must be deployed.

1. Right-click the desired CEP server to receive the SQL Query.
2. Select **Deploy Expert > Probes > SQL Query Monitor**.

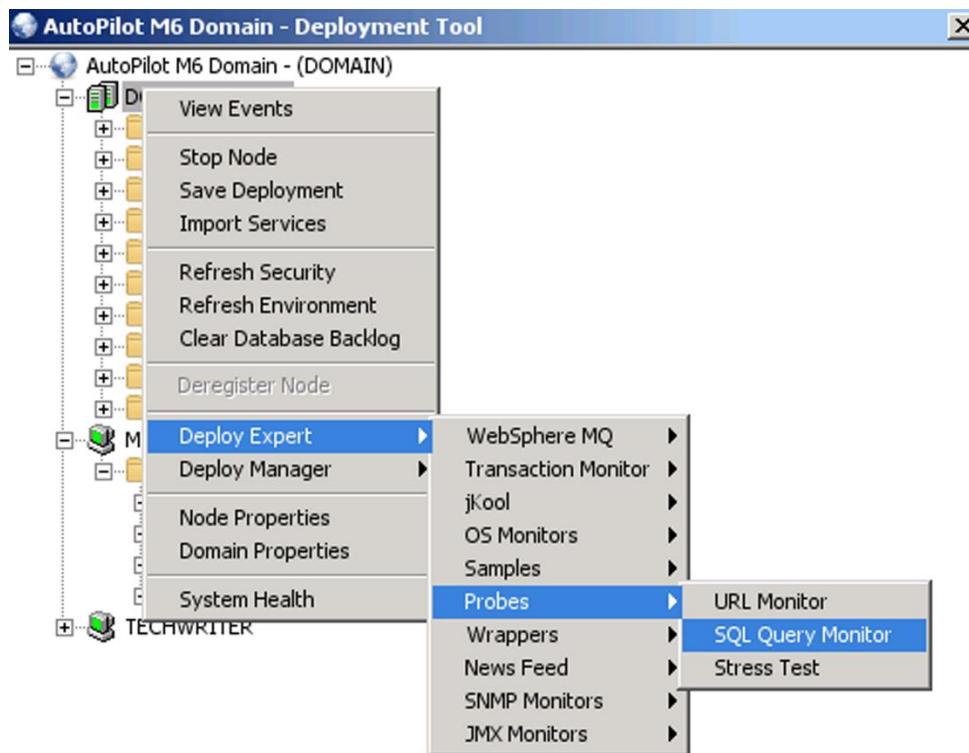


Figure 4-31. SQL Query Deployment

Configure as required for your application. An example for creating a query is shown below.

Creating an SQL Query Example:

After the SQL Query Monitor expert is deployed, do the following:

1. Give your query a name by selecting the *General* tab, if not already selected. The name must be unique within the meshIQ domain.

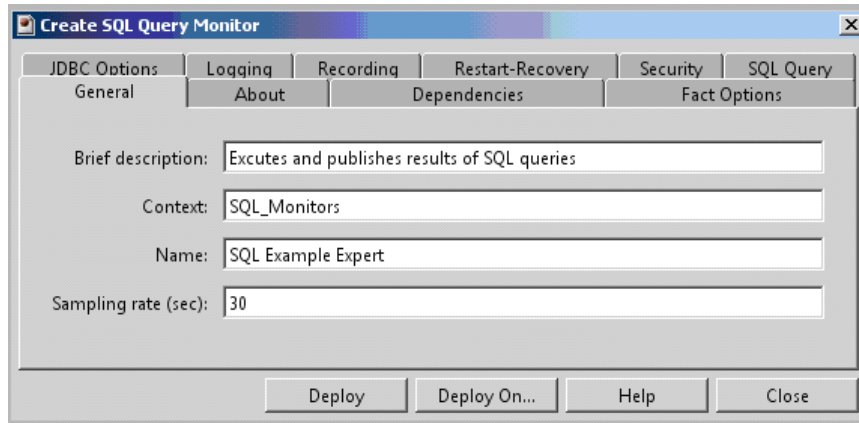


Figure 4-32. SQL Query, General Tab

2. To specify the required database, the credentials to access the database and the required driver, edit JDBC options by selecting the *JDBC Options* tab. This information will vary from database to database and from platform to platform.

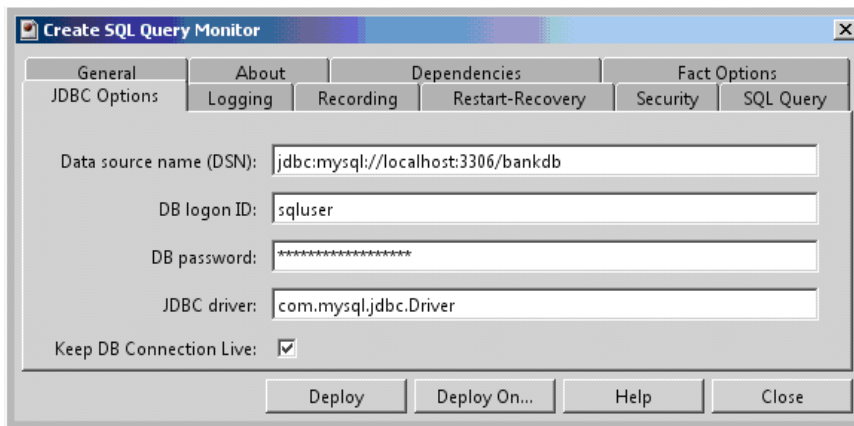


Figure 4-33. SQL Query, JDBC Options Tab

- To specify the query and structure for the facts, select the *SQL Query* tab.

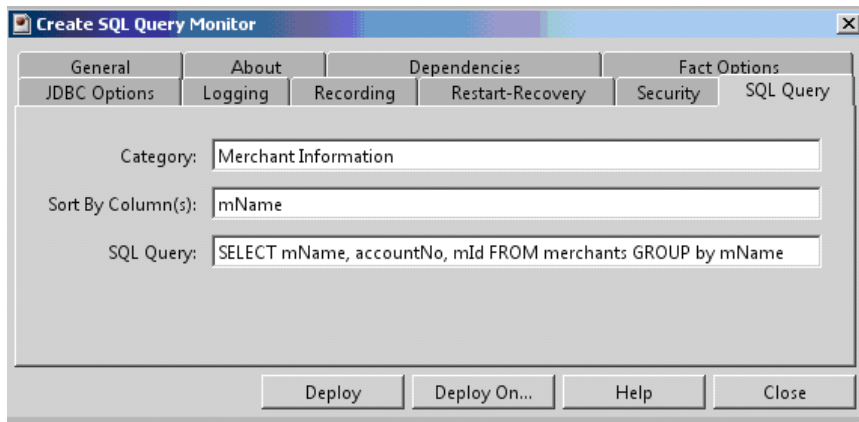


Figure 4-34. SQL Query, SQL Tab

In order to properly create facts, the query should return one row for each set of facts to be created in Core Services. In the figure below, the records are grouped and sorted by merchant name. Core Services is populated with the facts represented by the query. The facts are created under a category named *Merchant Info* which was specified on the *SQL Query* tab.

The **Sort By Column(s)** field is used to create three sets of facts for each merchant. The three facts selected were created as facts within each merchant.

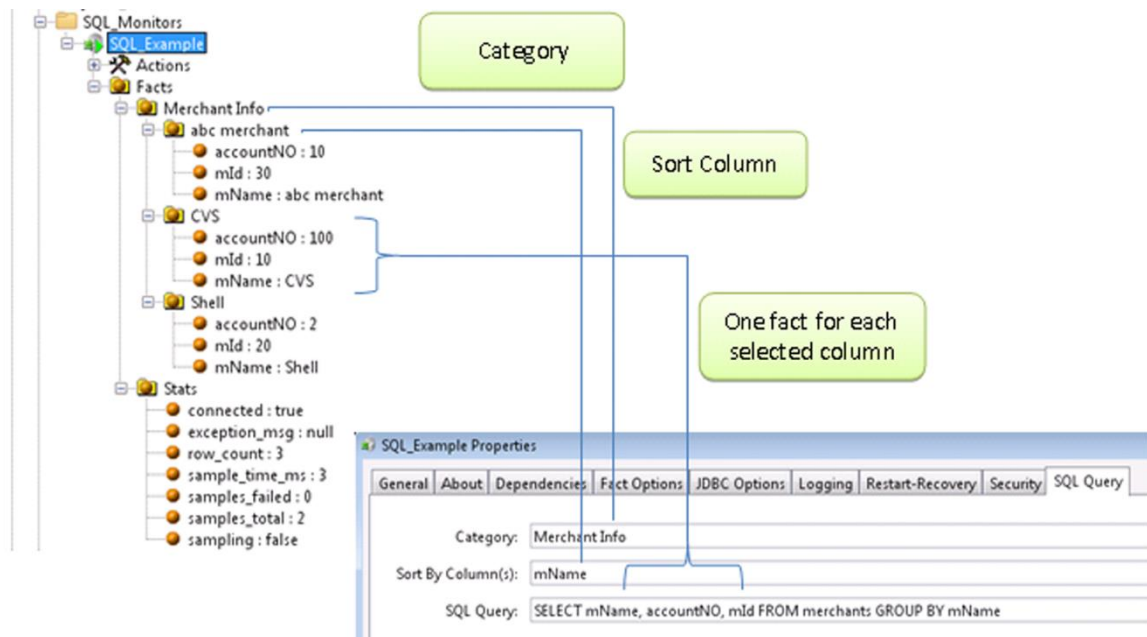


Figure 4-35. SQL Query Facts

4.5.1.3 RSS News Feed

meshIQ Platform Core Services has one built-in News Feed that lets you display current news based on the RSS URLs you selected to be scanned. This option is only available when the AIM plug-in is installed.

4.5.1.4 Wrappers

Two wrapper experts are built in as application templates. The wrappers are defined in the following paragraph.

- **Process Wrapper:** Allows the user to associate a process or a script that would be started by Core Services and be monitored by the process wrapper. Each wrapper may be associated with only one process or log. Process Wrapper is also used in conjunction with Fact Publishing utility **apfact**, which allows users to instrument user applications and publish metrics directly to the meshIQ Platform Core Services via a process wrapper.
- **File Monitor:** Separates log metrics from folder metrics. Lets you monitor the contents of a file in a specified directory. All folder-related facts are published under Facts\Folders. Also, lets you monitor application error logs and publish logs to Core Services as facts and events. All log-related facts are published under Facts\Logs. There is also support for multiple files and event profiles within a single file monitor instance.

4.5.2 Deploying Experts

Experts can be deployed on CEP servers (including the domain server) within the meshIQ Network. meshIQ Platform Core Services is supplied with built-in experts that can be used immediately after installation. In addition, several plug-ins are available which offer experts that are unique to each managed application.

This section provides you with instructions for the deployment of experts within the meshIQ network. The actual deployment of built-in experts should take just a few seconds in most cases. The following section is a typical example for deploying experts.

4.5.3 Deploying Process Wrapper

To deploy and configure an instance of a process wrapper:

1. Right-click the desired CEP server.
2. Select Deploy Expert > Wrappers > Process Wrapper
3. [Configure Process Wrapper](#) as required for your application.

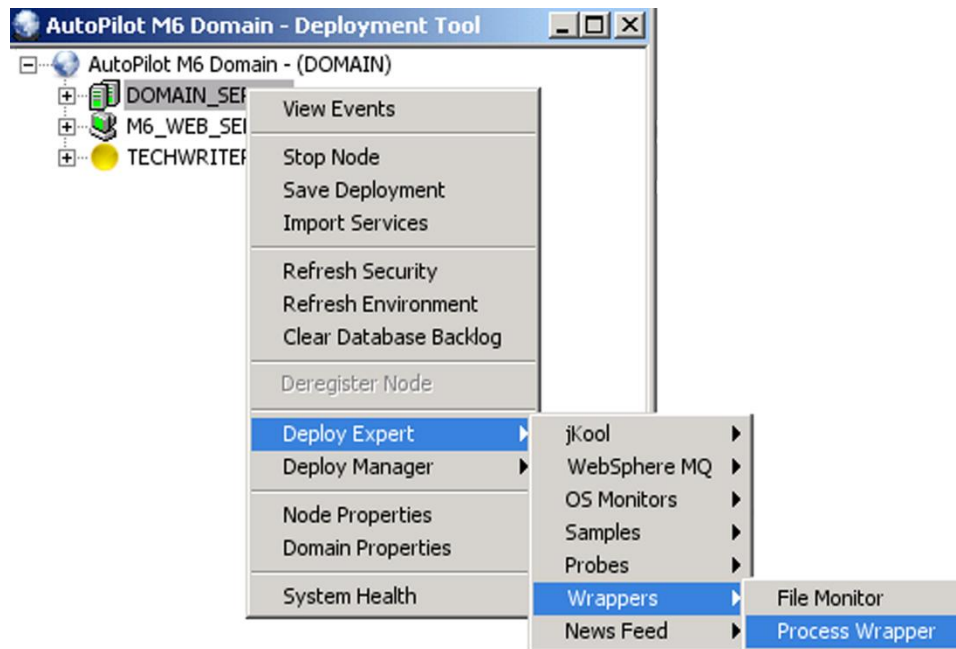


Figure 4-36. Deploying Process Wrapper Experts



The system assigned unique name is *Service* and timestamp in milliseconds. The name can be user defined but has to be unique within the meshIQ domain.

4. Click **Deploy** to deploy the expert on the CEP server. The *Create Process Wrapper* screen is displayed. The name of the expert being deployed is repeated along with the node where it will be deployed.

OR

Click **Deploy On** to deploy on multiple CEP servers. The *Deploy Across Network* screen is displayed. Select the nodes to receive the expert. You can use **Ctrl/click** to select multiple nodes in the domain. Click Deploy to deploy the expert on the selected nodes. When you have deployed, a check mark indicates which nodes have the expert deployed.

5. Right-click the CEP server(s) on which the expert was deployed and click **Save Deployment**.

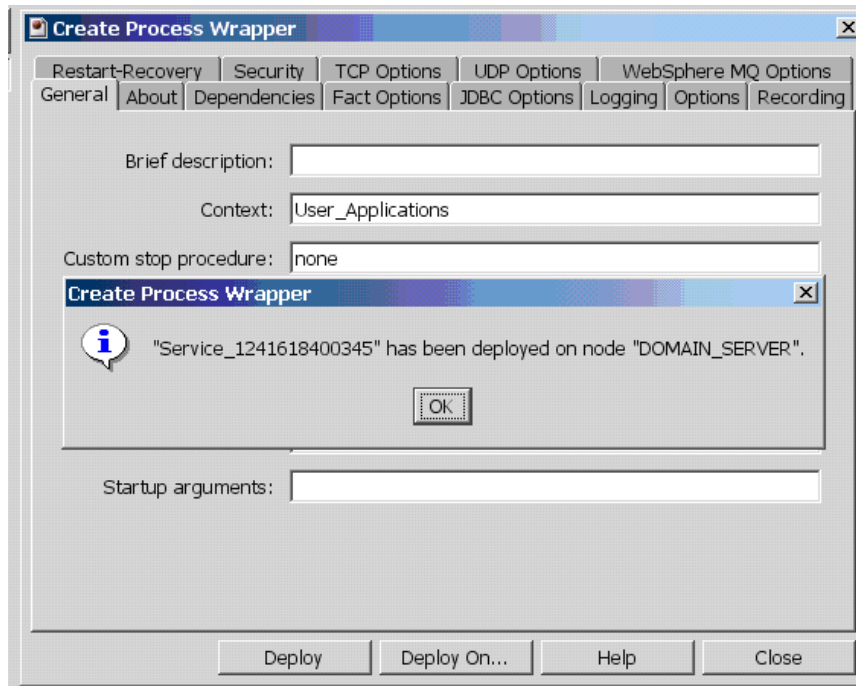


Figure 4-37. Deploying Wrappers

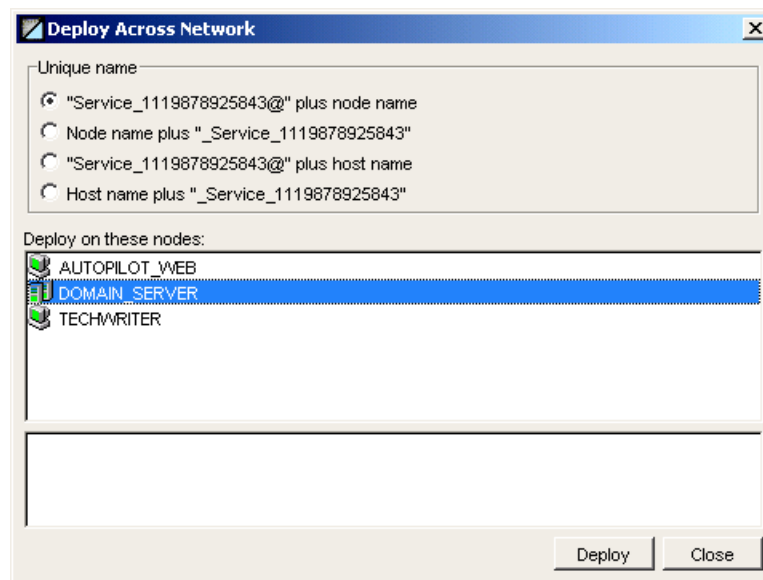



Figure 4-38. Deploy Wrapper on Multiple Nodes

4.5.4 Configuring Process Wrapper

A Process Wrapper can be used to initiate applications, accept facts from *apfact* and IBM MQ queues and record facts into a JDBC database.

Configure UDP Options section if process wrapper is used in conjunction with *apfact* (see [C.3 APFACT – Fact Publisher](#)).

A Process wrapper can also accept facts from an IBM MQ queue. Configure WebSphere MQ (IBM MQ) Options if facts are published into an IBM MQ queue. Applications that record facts into an IBM MQ queue must put messages with MQSTR format and message body should contain facts as described in [C.3 APFACT – Fact Publisher](#).



Process Wrapper uses IBM MQ Java client to communicate with the queue manager. Make sure that your queue manager is configured to accept IBM MQ clients – the queue manager must have a channel listener as well as a server connection channel defined.

Configure the process wrapper to support your needs using the table below. Mandatory or minimum entries are in **red**. There are 15 configuration screens for the process wrapper. Not all are required, but all are defined in the table.

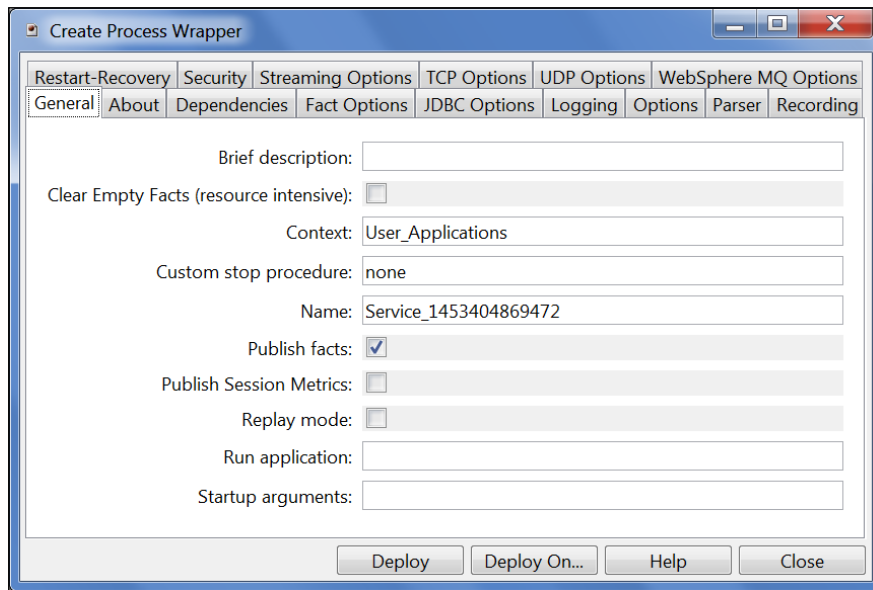


Figure 4-39. Process Wrapper General Properties

Table 4-16. Process Wrapper Expert Configuration	
Property	Description
General	
Brief Description	A short, user-defined description of the service
Clear Empty Facts	When enabled, clears facts with empty or null values.
Context	A user-defined category that will be registered with the domain server. The default is: <i>User_Applications</i> . Context can be changed as needed. Any new user-defined Context value that you enter is used to name a new folder in Enterprise Manager.
Custom stop procedure	Application or script that gracefully shuts down started applications. The default is: <i>none</i> . The application will be terminated using the OS specific stop procedure.

Table 4-16. Process Wrapper Expert Configuration

Property	Description
Name	Name that uniquely identifies the service in the domain. The default naming system uses the word <i>service</i> followed by 12 random digits (example: service_123456789012)
Publish facts	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to publish received facts locally.
Publish Session Metrics	When enabled, publishes session performance metrics.
Replay Mode	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to emulate fact source and replay all received facts.
Run Application	Fully qualified name of application/script/process to run (example: /usr/bin/tar)
Startup Arguments	Startup parameters that are passed to the application (example: cvf mytar/opt/meshiq)
About	
Package title	Implementation title of the source package.
Package vendor	Name of implementation vendor.
Package version	Package version as assigned by the vendor.
Dependencies	
Platform Dependencies	Comma-separated list of operating system platforms this expert is dependent on.
Service Dependencies	Comma-separated list of services this expert is dependent on.
Fact Options	
Exclude Expire Filter (regexp)	Do not expire facts that match specified regular expression.
Exclude Fact Filters	Comma-separated list of fact paths to exclude during publishing.
Expire facts (ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size *	Automatically maintains specified number of samples for each published fact in memory
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Fact service alias	Override fact service prefix for all published facts. Facts will be appearing under specified service name.
Include Expire Filter (regexp)	Facts that match the specified regular expression are expired
Include Fact Filters	Comma-separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to Fact History Time or Fact History Size , whichever limit is reached first. If disabled, newer history samples replace older ones on a rolling basis.
JDBC Options	

Table 4-16. Process Wrapper Expert Configuration

Property	Description
Data source name (DSN)	Logical data source name that points to the physical database.
DB logon ID	Login ID to access database if required.
DB password	Password to access database if required.
DB table	Name of the physical database table (example: CPU_TABLE)
Enable DB logging	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> database logging.
JDBC driver	Class name of the JDBC driver.
User table columns	(LogTime) VALUES (?)
Logging	
Audit	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service audit trace.
Log name	Log name associated with the service.
Log service activity	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service activity trace. (Should only be enabled for troubleshooting purposes to eliminate significant performance degradation.)
Log size (bytes)	Log size in bytes. Real log size is the maximum value of server.log.size and logsize.
Options	
Auto-restart delay (ms)	Delay before restarting stopped applications.
Auto-restart	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to specify whether application should be restarted when stopped.
Environmental variables	Comma-separated list of environment variables required for the application (example: PATH=/opt.meshiq,TEMP=/temp)
Is daemon	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to specify whether application is long running (daemon) or terminates after execution.
Redirect output	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to redirect program output to private event log.
Parser	
Fact name group	Obtain fact name part from the specified group in the fact pattern expression.
Fact pattern	Fact pattern expression used for parsing incoming facts.
Fact value group	Obtain value part from the specified group in the fact pattern expression.
Quote character	Quote character used to parse incoming facts; default is " (double quotation mark)
Space characters	Space characters used to parse incoming tokens (default is , (comma) and ; (semicolon))
Recording	

Table 4-16. Process Wrapper Expert Configuration

Property	Description
Anomaly Deviation Limit	Number of standard deviations above or below the mean.
Exclude Filter (regex)	A regular expression filter to exclude certain facts from being written to the database. Facts have the format <code>expert\class\instance\leaf=value</code> such as in the example <code>Servers\Linux\Serv7\processes=40</code> .
Fact Anomaly Frequency	Frequency at which anomalies are checked and recorded.
Fact State Frequency	If Record Fact State is enabled, the value entered here specifies how often the Fact State is updated.
Fact Summary Frequency	If Record Fact Summary is enabled, used to write an intermediate summary record every X th update to the fact during the Summary Interval. This is done to avoid waiting the full Summary Interval for a summary record to appear in the summary table.
Include Filter (regex)	A regular expression filter to include certain facts being written to the database. Same format as described for the exclude filter.
Record Fact Anomalies	Enable/disable fact anomaly recording for this service.
Record Fact History	If enabled, records every fact change into the History database. The exclude/include filters are respected. To define database tables and set Core Services options, refer to Recording Facts .
Record Fact State	If enabled, records the last value published (current state) into the state database and restores that value when the CEP Server is stopped and restarted. The exclude/include filters are respected. To define database tables and set Core Services options, refer to Recording Facts .
Record Fact Summary	If enabled, records summary record at the interval designated in the Summary Interval (ms) field into the Summary database. The exclude/include filters are respected. To define database tables and set Core Services options, refer to Recording Facts .
Storage for Anomalies	SQL table where anomalies are recorded.
Storage for History	Database table where the Fact History data is stored.
Storage for State	Database table where the Fact State data is stored.
Storage for Summary	Database table where the Fact Summary data is stored.
Summary Interval (ms)	If Record Fact Summary is enabled, designates how often the Fact Summary data is written (in milliseconds).
Restart-Recovery	
Automatic start	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> automatic restart.
Save in registry	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to save persistent service in registry.
Synchronous Control	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to initiate synchronous service.

Table 4-16. Process Wrapper Expert Configuration

Property	Description	
Security		
Inherit permissions from owner	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> inheriting of permission from owner's permission masks.	
Owner	User that owns the object.	
Permissions	Permissions for users of the same group and others. Disable/Enable as required.	
	Group:	Other Users:
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, and disable.	Others may execute control actions such as start, stop, and disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.
Streaming Options (Refer to Streaming Data.)		
Application name	Sets application name	
Data center name	Sets data center name	
Exclude filter (regex)	Ignore facts that match specified regular expression	
Include filter (regex)	Log facts that match specified regular expression	
Location	Sets server location	
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework)	
Streaming configuration	Streaming configuration block name	
TCP Options		
Accept TCP facts	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to accept facts published to the specified TCP port.	
Option SO_TIMEOUT	TCP option - read timeout	
TCP port	Unique port on which this service will listen on for incoming facts	
UDP Options		
Accept UDP facts	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to accept facts published to the specified UDP port.	
UDP port	Unique port on which this service will listen on for incoming facts.	
WebSphere MQ (IBM MQ) Options		

Table 4-16. Process Wrapper Expert Configuration

Property	Description
Accept truncated msg	Accept and truncate incoming message if exceeds maximum message size.
Facts from WebSphere MQ	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> reading facts from IBM MQ.
Log message context	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to log IBM MQ descriptors such as type and persistence.
Password	Password to be used when connecting to the queue manager.
Queue manager host	Name of the host where the queue manager is defined.
Queue manager name	Name of the local queue manager.
Queue manager port	Channel listener TCP port for the queue manager.
Queue name	Name of the queue where the facts are being published.
Server connection channel	Name of the SVR CONN channel on the queue manager.
SSL certificate store	SSL certificate store location
SSL cipher suite	Cipher suite that matches the CipherSpec of the channel
SSL enable	Enable/disable <input checked="" type="checkbox"/> / <input type="checkbox"/> communication for channel communication.
User ID	User ID to be used when connecting to the queue manager.

* **Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to [Fact Volatility](#) for more detailed information on fact volatility.)

4.5.4.1 Recording Facts

To enable fact recording for this service, click the *Recording* tab (Figure 4-40). You can filter the facts you want to record by using the exclude/include filters.

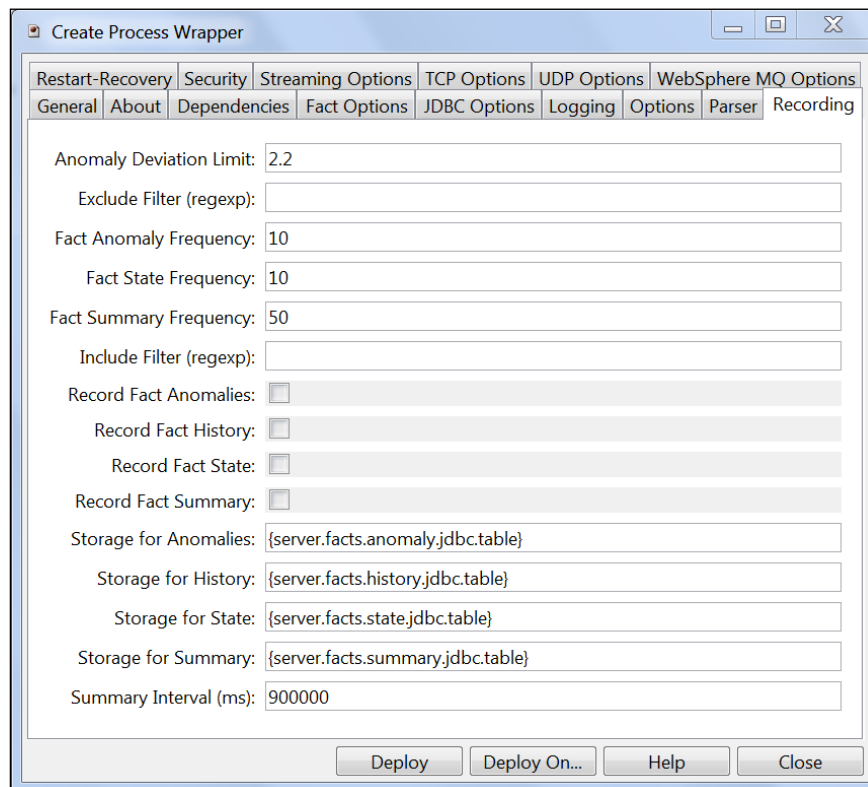


Figure 4-40. Recording Characteristics

Refer to Table 4-16 for an explanation of each *Recording* property. In addition, to enable fact recording for state, history, or summary, you must complete the following steps to define the database tables and set the Core Services options. The following steps are for MySQL.

1. Create a database schema, for example `history_logging`.
2. Use the sample SQL in `[AUTOPILOT_HOME]/sql-scripts/system/core-mysql.sql` to create the tables required.
3. Copy the sample properties from `[AUTOPILOT_HOME]/sql-scripts/system/core-sql.properties` into the require properties file, for example `[AUTOPILOT_HOME]/localhost/node.properties`. (See [Server Runtime - PROPERTY FILES](#) for more information.)
 - a. Change the following properties to match your installation:
 - property `server.service.facts.logging=true` - must be added to activate fact recording (for testing can be specified as `record` when starting the server on the command line)
 - property `server.facts.jdbc.driver=com.mysql.jdbc.Driver` - specifies the name of the JDBC driver

- property `server.facts.url=jdbc:mysql://localhost:3306/history_logging` - specifies the url (host and port) and schema name for the MySQL Database
 - property `server.facts.jdbc.user=user` - specifies the user to use to connect
 - property `server.facts.jdbc.password=pswd` - specifies the password to use to connect
 - property `server.facts.summary.jdbc.table=ft_summary` - default value for the summary table (can be overridden as shown above)
 - property `server.facts.state.jdbc.table=ft_state` - default value for the state table (can be overridden as shown above)
 - property `server.facts.history.jdbc.table=ft_history` - default value for the history table (can be overridden as shown above)
- b. Review the following and change only as needed:
- property `server.facts.summary.update.frequency=50` - frequency which summary records are updated (a lower value increases the potential I/O activity, a higher value increases the time before recent data is written to summary table)
 - property `server.facts.state.update.frequency=10` - default frequency at which state records are synced to the database. A value of 1 writes the state information to the database every time the fact is published and should be used for low volume facts.
 - property `server.facts.jdbc.batch.size=20` - JDBC batch size used for writing fact history
 - property `server.facts.jdbc.retry=30000` - JDBC connection retry in ms. in case of a SQL failure

4.5.4.2 Process Wrapper JDBC Configuration

This section describes the procedure for enabling database fact logging with a Process Wrapper. There are several important steps that must be taken to ensure proper database insertion. These steps include determining the source of the data, configuring the insertion format, and configuring the database table correctly.



The functionality provided by using JDBC options can be done much simply by using the *Recording* tab and selecting the facts to record.

Determining Data Source

There are several ways that a Process Wrapper can receive data and publish as facts. The first method is using TCP protocol, the second UDP and the last is from an MQ queue.

TCP

TCP is a reliable data connection to ensure facts will be published, but there is a slight performance hit due to the extra networking overhead required.

1. Click *TCP Options* tab, to enable a Process Wrapper to receive TCP data.
2. Check *Accept TCP Facts* checkbox , and then enter a port. The port will also have to be specified in the application sending the fact data.

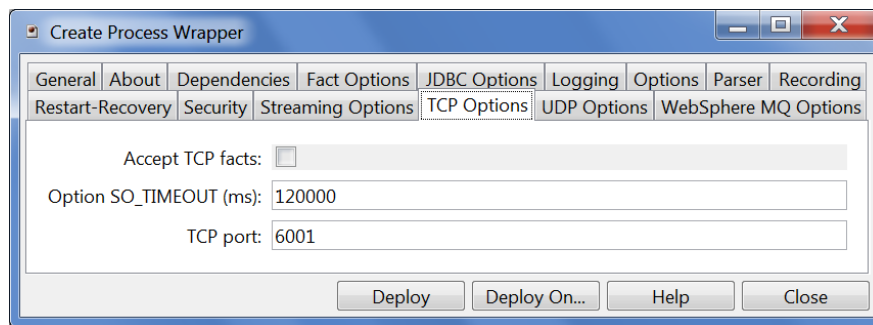


Figure 4-41. TCP Options for JDBC Configuration

UDP

UDP is an unreliable data connection used mainly for speed and application-to-application decoupling. While this is the fastest protocol with the least amount of network overhead, there is potential that data could be lost since there is no acknowledgment/hand shaking between the sender and receiver.

1. Click *UDP Options* tab to enable a process wrapper to receive UDP data.
2. Check *Accept UDP Facts* checkbox and then enter a port. The port will also have to be specified in the application sending the fact data.

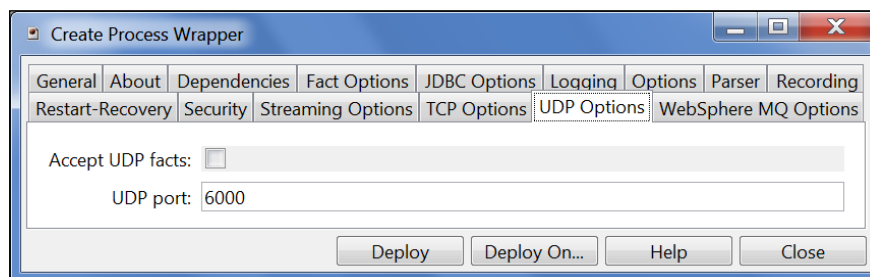


Figure 4-42. UDP Options for JDBC Configuration

WebSphere MQ (IBM MQ)

IBM MQ is an asynchronous assured delivery messaging platform. This protocol is used when messages cannot be lost and must be decoupled from publishing applications.

1. Click the *WebSphere MQ Options* tab (for IBM MQ) to enable a Process Wrapper to receive MQ message data.
2. Check the *Facts from WebSphere MQ* checkbox.
3. Configure connection to the Queue Manager that will receive the message data from the source.
4. Enter Queue Manager Hostname.

5. Enter Queue Manager name (case sensitive).
6. Enter listener port and the Queue name where the messages will be taken.
7. Enter Server connection channel name.
8. Enter user ID.
9. Enter password if specified in the channel definition.

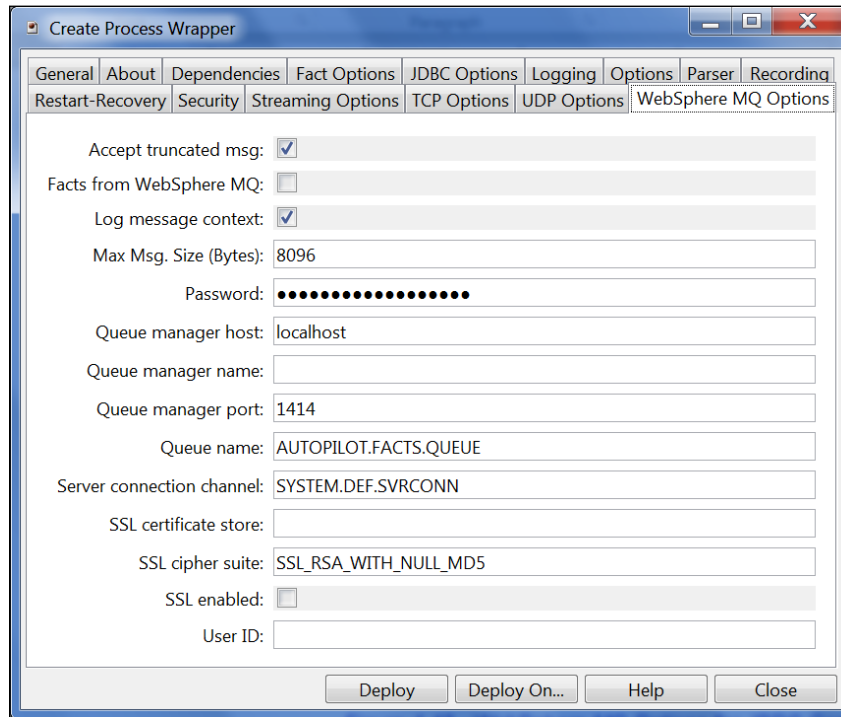


Figure 4-43. WebSphere MQ (IBM MQ) Options for JDBC Configuration

The *Log Message Context* will log additional information about each message that is received from the MQ Queue. This data captured from each message is:

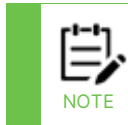
**messageType, persistence, priority, characterSet, encoding,
putApplicationType, replyToQueueManagerName, replyToQueueName,
applicationIdData, applicationOriginData, TotalMessageLength, ptime,
messagId, correlationId, accountingToken**

In the data field of the MQ messages being placed on the MQ Queue by the sending application, data must be paired by **fact_name=value** and comma separated. For example:

```
FLOW=NASHUB_FLOW, EVENT=FLOWTIME, TOTALTIME_TIMEMETRIC = 1308,  
NASTIME_TIMEMETRIC = 1128, TOTALNASTIME_TIMEMETRIC = 1128, NAST_UID = 39196  
WWS0330300007001NN1245448920050303, MSGID =  
414d5120574249514130312020202020421e8db0201836b8, REINPUTS = 0, PUTTIME =  
17:45:48.350, NAS_TRANID = CISI50
```

The facts will be overwritten every time a new message is written to the queue and retrieved by the Process Wrapper. These facts can be used to define business rules with thresholds. They could also be correlated together or with other metrics published within other parts of Core Services.

Configuring JDBC Options:



See the JDBC documentation for the requirements for your database type.

Once the data is obtained and written into meshIQ Platform Core Services, this same data can be written to a database. The reasons for doing this may be to retain critical corporate information for historical purposes, capacity planning, or compliance with industry business standards such as Sarbanes-Oxley.

1. Click *JDBC Options* tab to enable a process wrapper to write fact data to a database.
2. The Data Source Name (DSN) and the JDBC Driver will be the following format:

Table 4-17. Process Wrapper JDBC Options Configuration

Database	JDBC Driver	Data Source Name (DSN) format
ODBC Compliant	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:dsn_name
MS SQL Server (lib/jtds.jar)	net.sourceforge.jtds.jdbc.Driver	jdbc:jtds:sqlserver://host:port/dbname
Sybase (lib/jtds.jar)	net.sourceforge.jtds.jdbc.Driver	jdbc:jtds:sybase://host:port/dbname
Oracle (lib/ojdbc14.jar)	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@hostname:port:sid
DB2 UDB (lib/db2java.zip, lib/db2jcc.jar)	COM.ibm.db2.jdbc.net.DB2Driver	jdbc:db2://hostname:port/dbname
Hypersonic SQL	org.hsql.jdbcDriver	jdbc:hypersonicSQL:hsql://host:port/dbname
Informix	com.Informix.jdbc.IfxDriver	jdbc:informix-sqli://host:port/dbname
MYSQL	com.mysql.jdbc.Driver	jdbc:mysql://host:port/dbname

3. Enter user ID and password.
4. Enter the name of the database table.
5. Check *Enable DB Logging* checkbox.
6. Configure the connection to the database that will receive the data.
7. Set up the *User table columns* properties: This is the most critical part to ensure proper database table insertions.
 - You must know the **order** that the fact data is being published into the process wrapper. This is not initially intuitive because once the facts are received, they are viewed through the User Console in alphabetical order. To determine the order facts are received, an examination of the sending source must be done.
 - In the case of UDP or TCP, the script or application will have to be viewed to make the determination of the order of publishing. It should be a simple line by line or loop.

- In the case of MQ, the MQ application can be examined, or a simpler approach would be to look at one of the messages using Message Explorer. Once the order is determined, the *User table columns* can be defined correctly.
- If the order is not accounted for, data could be inserted into the wrong columns in the database.

The format for this field follows this example:

```
(LogTime, FLOW, EVENT, TOTALTIME_TIMEMETRIC, NASTIME_TIMEMETRIC,
TOTALNASTIME_TIMEMETRIC, NAST_UID, MSGID, REINPUTS, PUTTIME, NAS_TRANID)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
```

Note that:

- LogTime is a prerequisite for any configuration and is **always** the first field in the user table columns string. This field is defined by default when a Process Wrapper is created.
- After LogTime, insert all the columns in the correct order as they are published. Make sure the syntax is correct. Characters are case sensitive. The entire list is comma delimited and ended with a parenthesis.
- Next, VALUES are defined. This is simply used by the application to know how many columns it has to publish. The total number of question marks should match the total number of columns to be published. Simply count each one, including LogTime, and place the correct number of question marks in the VALUE field.

The following figure is an example of a configured Process Wrapper to publish fact data to a database:

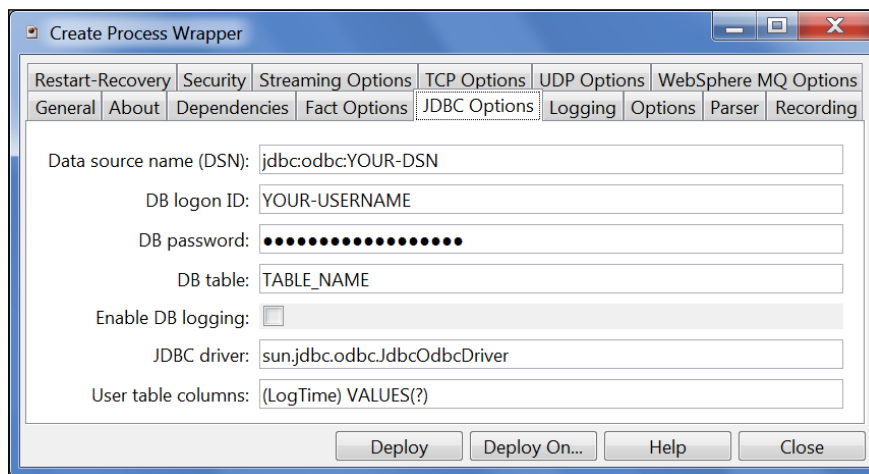


Figure 4-44. User table columns of JDBC Process Wrapper

Configuring the Database:

The final step to ensure proper database logging is to correctly define the database objects. In this case, this means the database table and columns.

- The database can be given a meaningful name, but it is user defined.

- Table column names should be defined the same way as they are in the *User table columns* in the JDBC Options configuration.
- To determine what data type each column should be, the Database Tool can be used. To use the Database Tool, make sure your User Console is open and click the Tools menu and choose Database Tool. Once opened, it will look like this example:

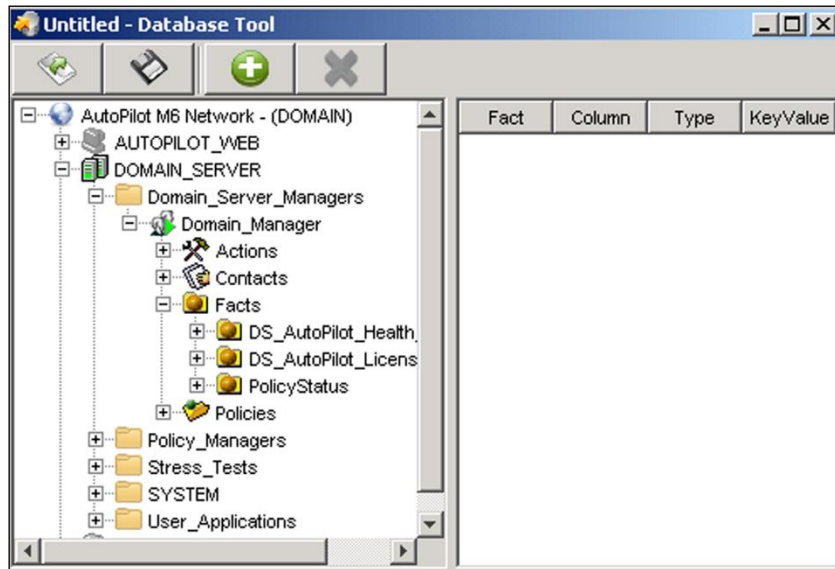



Figure 4-45. User Console Database Tool

1. Navigate to the facts being published by the process wrapper:
2. Add each of the facts to the scheme template by selecting the fact and clicking the  icon in the toolbar. When the fact is added the data type of each fact will be displayed in the *Type* column.

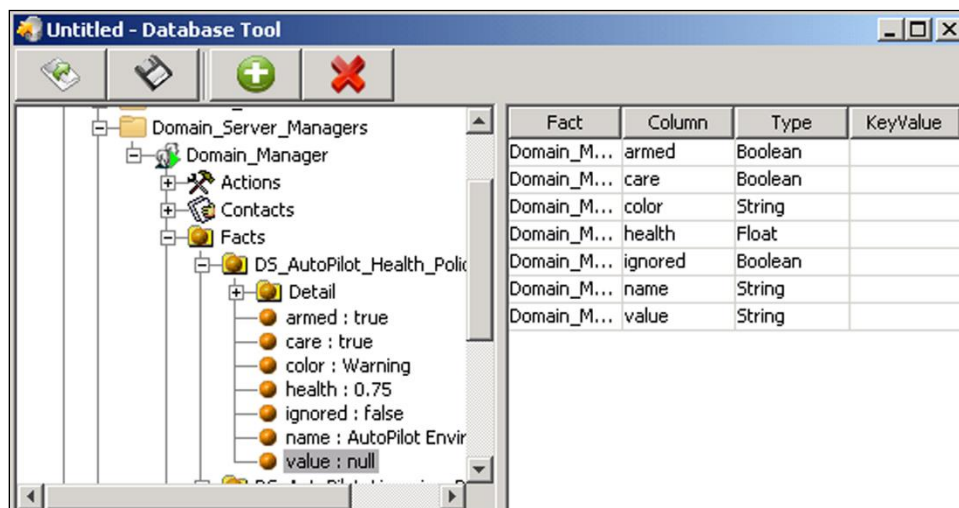


Figure 4-46. Adding Published Facts to Database Scheme

3. From this view, define your database data types for each column. Ensure the database is available and the user ID has the correct rights to insert to the database.

Verification:

Once the configuration is complete from the publishing side, Core Services side and database side, start the process wrapper and observe if facts are being published. Depending on the application sending the facts, they may only come at specified intervals. Once facts are populated, we know that data is being received correctly. In turn, logging to database should happen as soon as the first batch of facts is received. To verify the data is going into the database, the user can use a DB admin tool or SQL query to view/edit table contents. If there are no entries in the database, then the user should check the log files. Mainly, right-click the Process Wrapper and select *View Events*. If there are connection or insertion problems, they will be reported in this log file. Typically, connection problems are caused by syntax errors made in setting up the JDBC *User table columns* and/or the database column name/data type definitions.

4.5.5 Deploying File Monitors

To deploy and configure an instance of a file monitor:

1. Right-click the desired CEP server.
2. Select **Deploy Expert > Wrappers > File Monitor**
3. Configure the file monitor as required for your application.

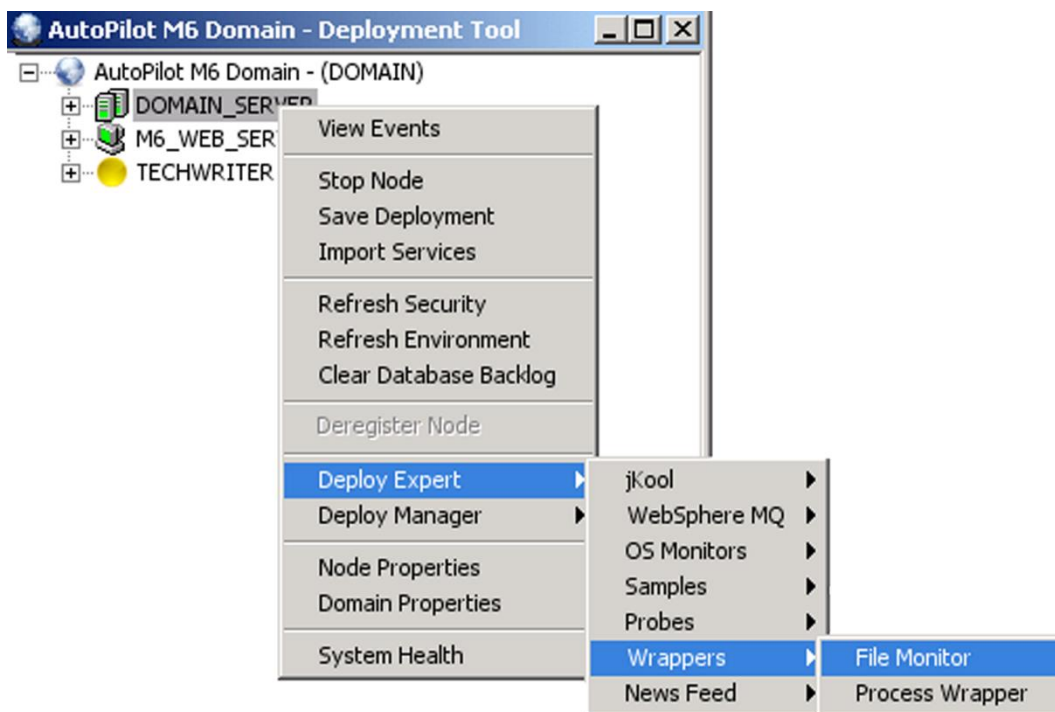


Figure 4-47. Deploying File Monitors



NOTE

The system-assigned unique name is *Service* and timestamp in milliseconds. The name can be user-defined but has to be unique within the meshIQ domain.

- Click **Deploy** to deploy the expert on the CEP server. The *Create File Monitor* screen is displayed. The name of the expert being deployed is repeated along with the node where it will be deployed.

OR

Click **Deploy On** to deploy on multiple CEP servers. The *Deploy Across Network* screen is displayed. Select the nodes to receive the expert. You can use **Ctrl/click** to select multiple nodes in the domain. Click **Deploy** to deploy the expert on the selected nodes. A check mark indicates which nodes have the expert deployed.

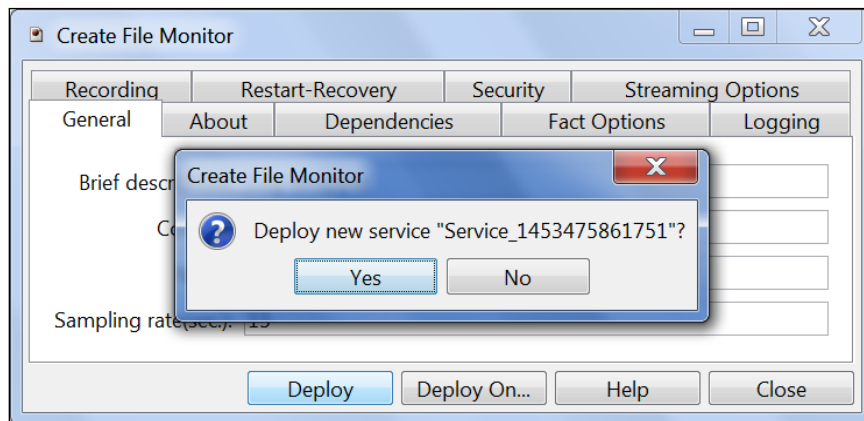


Figure 4-48. Deploying File Monitor on Individual Node

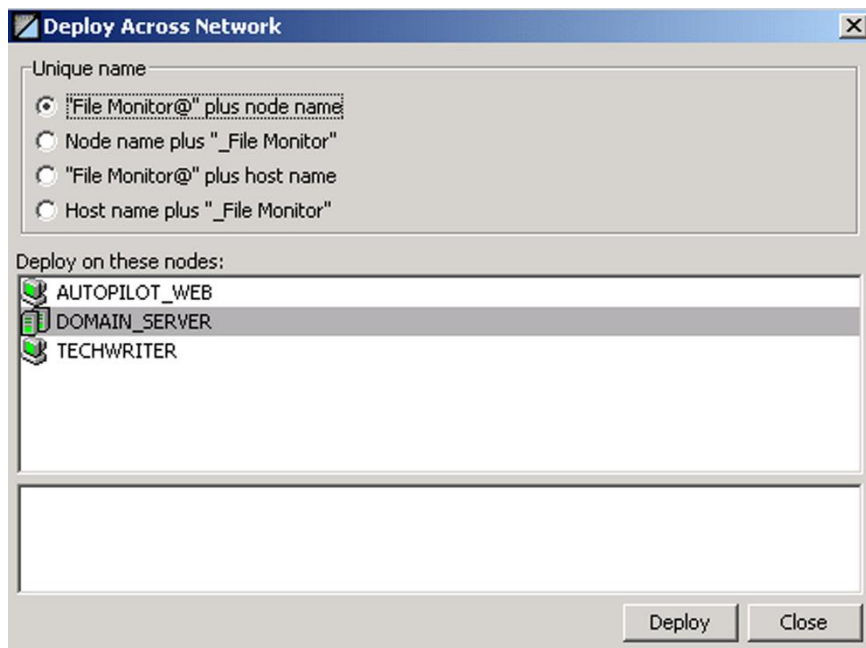


Figure 4-49. Deploying File Monitor on Multiple Nodes

4.5.5.1 Configuring File Monitors

Configure the File Monitor to support your needs using the table below. Mandatory or minimum entries are in **red**. There are eight configuration screens for the File Monitor. Not all are required, but all are defined in the table.

Table 4-18. File Monitor Expert Configuration	
Property	Description
General	
Brief Description	A short, user-defined, description of the service.
Context	A user-defined category that will be registered with the domain server. The default is: <i>User_Applications</i> . The context can be changed as needed. Any new user-defined Context value that you enter is used to name a new folder in Enterprise Manager.
Name	Name that uniquely identifies the service in the domain. The default naming uses the word <i>service</i> followed by 13 random digits (example: <i>Service_1234567890123</i>)
Sampling Rate (sec)	How many times per second the fact is sampled.
About	
Package Title	Implementation title of the source package.
Package vendor	Name of implementation vendor.
Package version	Package version as assigned by the vendor.
Dependencies	
Platform Dependencies	Comma-separated list of operating system platforms this expert is dependent on.
Service Dependencies	Comma-separated list of services this expert is dependent on.
Fact Options	
Exclude Expire Filter (regexp)	Do not expire facts that match the regexp
Exclude Fact Filters	Comma-separated list of fact paths to exclude during publishing.
Expire facts (ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size*	Automatically maintains the specified number of samples for each published fact in memory.
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Fact service alias	Override fact service prefix for all published facts. Facts will appear under specified service name.

Table 4-18. File Monitor Expert Configuration

Property	Description
Include Expire Filter (regex)	Expire facts that match the regex.
Include Fact Filters	Comma-separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to Fact History Time or Fact History Size , whichever limit is reached first. If disabled, newer history samples replace older ones on a rolling basis.
Logging	
Audit	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service audit trace.
Log name	Log name associated with the service.
Log service activity	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service activity trace.
Log size (bytes)	Log size in bytes. Real log size is the maximum value of server.log.size and logsize.
Recording	
Anomaly Deviation Limit	Number of standard deviations above or below the mean.
Exclude Filter (regex)	A regular expression filter to exclude certain facts from being written to the database. Facts have the format <code>expert\class\instance\leaf=value</code> such as in the example <code>Servers\Linux\Serv7\processes=40</code> .
Fact Anomaly Frequency	Frequency at which anomalies are checked and recorded.
Fact State Frequency	If Record Fact State is enabled, the value entered here specifies how often the Fact State is updated.
Fact Summary Frequency	If Record Fact Summary is enabled, used to write an intermediate summary record every X th update to the fact during the Summary Interval. This is done to avoid waiting the full Summary Interval for a summary record to appear in the summary table.
Include Filter (regex)	A regular expression filter to include certain facts being written to the database. Same format as described for the exclude filter.
Record Fact Anomalies	Enable/disable fact anomaly recording for this service.
Record Fact History	If enabled, records every fact change into the History database. The exclude/include filters are respected. To define database tables and set options, refer to Recording Facts .
Record Fact State	If enabled, records the last value published (current state) into the state database and restores that value when the CEP Server is stopped and restarted. The exclude/include filters are respected. To define database tables and set options, refer to Recording Facts .
Record Fact Summary	If enabled, records summary record at the interval designated in the Summary Interval (ms) field into the Summary database. The

Table 4-18. File Monitor Expert Configuration

Property	Description	
	exclude/include filters are respected. To define database tables and set options, refer to Recording Facts .	
Storage for Anomalies	SQL table where all anomalies are stored.	
Storage for History	Database table where the Fact History data is stored.	
Storage for State	Database table where the Fact State data is stored.	
Storage for Summary	Database table where the Fact Summary data is stored.	
Summary Interval (ms)	If Record Fact Summary is enabled, designates in milliseconds, how often the Fact Summary data is written.	
Restart/Recovery		
Automatic start	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> automatic restart.	
Save in registry	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to save persistent service in registry.	
Synchronous Control	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to initiate synchronous service.	
Security		
Inherit Permission from Owner	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> inheriting of permission from owners' permission masks.	
Owner	User that owns the object.	
Permissions:	Permissions for users of the same group and others. Disable/Enable as required.	
	Group:	Other Users:
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, and disable.	Others may execute control actions such as start, stop, and disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.
Streaming Options (Refer to Streaming Data.)		
Application name	Sets application name	
Data center name	Sets data center name	
Exclude filter (regexp)	Ignore facts that match specified regular expression	
Include filter (regexp)	Log facts that match specified regular expression	

Table 4-18. File Monitor Expert Configuration

Property	Description
Location	Sets server location
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework)
Streaming configuration	Streaming configuration block name

* **Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to [Fact Volatility](#) for more detailed information on fact volatility.)

4.5.5.2 Multiple Log File Support

A File Monitor can provide support for multiple files and events profiles within a single File Monitor instance. The File Monitor will publish the following additional facts:

- EventOccurence
- EventTime
- EventText
- History/EventOccurence
- History/EventText
- History/EventTime

This is accomplished only after the File Monitor has been deployed.

Add Log File and Event Profile:

1. Expand the deployed File Monitor.
2. Right-click the *Logs* folder and click **Add Log File**.
3. Enter Log File properties and click **OK**.

Log file – Path where log file is located.

Log file alias – User-defined name for log file.

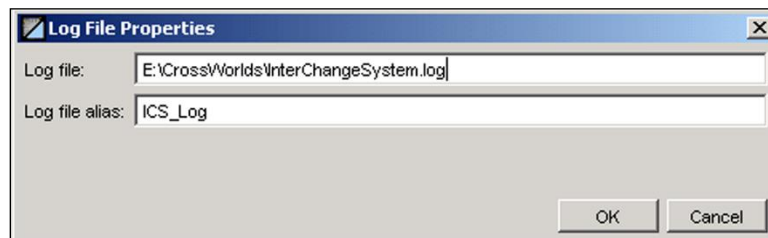



Figure 4-50. Adding Log Monitors

4. Expand the folder for the Logs file.
5. Right-click *Logs* folder and click **Add Event Profile**.
6. Configure the event profile as described below.


NOTE

An asterisk (*) in the Description field signifies no special requirement.

Table 4-19. Event Profile Properties	
Property	Description
General	
Event profile name	User-defined name for event.
Event ID	ID of the event to look for.
Event type	Specifies the severity of the event. Values are from 0 through 8.
History size	Number of event occurrences to keep in history. (1 = only the most recent occurrence is stored.)
Filters	
Event begins as	Specifies text the event must begin with. Typically specified as Error*
Event ends as	Specifies text the event must end with. Typically specified as *end
Event filter	Specifies text that must be contained within the event. Typically specified as *text* (*may occur more than once.)
Number of lines per event	Displays the event with the number of lines specified.

Example of History size: To detect an event occurrence of 10, type 10 in the History size text field on the Event Profile Properties screen. The Log Monitor will hold 10 entries in memory and publish the oldest entry under History. EventOccurrence will count up from 1 and the Update-Count can be checked for the number of event occurrences since last reset (Reset-Fact() action). Reset_Fact actions should be used to reset the Update-Count on the EventOccurrence.

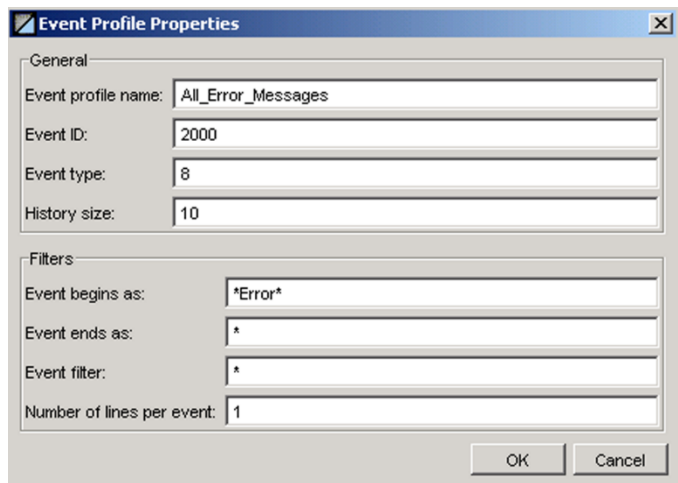


Figure 4-51. Event Profile

Filters Example:

The line

error this is a text message that I want that contains red and ends with blue
can be specified in two ways:

1. *Event begins as:* error*

Event ends as: *blue

Event filter: *red*

OR

2. *Event begins as:* *

Event ends as: *

Event filter: *red*blue

If 3 lines were specified as the *Number of lines per event*, then the result would be:

error

red

blue

Add New File Monitor Subfolder:

1. Expand the folders for the deployed File Monitor.
2. Right-click the *Folders* subfolder and click **New Folder Monitor**.
3. Enter Folder Monitor properties and click **OK**.

Table 4-20. Folder Monitor Properties

Property	Description
Folder alias	User-defined name for folder.
Folder name	Path where log file is located.
File filter	Directory of filter. For example: /var/x.log
Include sub-folders check box	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to include sub-folders.



NOTE

An asterisk (*) in the Description field signifies no special requirement.

Figure 4-52. Creating New Folder Monitor

Edit File Monitor:

1. Right-click File Monitor to be edited and click **Properties**.
2. Change File Monitor properties and click **Apply**.

Remove File Monitor:

1. Right-click File Monitor to be removed and click **Remove**.
2. A Confirmation window will be displayed. Click **yes** to remove the File Monitor.

4.5.6 Deploying IBM MQ Experts

See the Resource Center for documentation on how to configure and deploy IBM MQ experts.

4.6 Managers

The operational structure within Core Services is similar to a human management structure. A manager is a management service that has authority, responsibility, delegation, and decision-making ability. Managers have the following characteristics:

- Mobile software components equipped with policies and contacts.
- Filter information before forwarding up the hierarchy to other managers.
- Managers behave similarly to experts, except managers are not specific to a managed resource and can be equipped with advanced automation policies and functions.
- Managers can be made aware of, and manage, any application through the subscribing experts.
- Managers communicate with each other through policies and facts.
- Managers can be created on any CEP server from the User Console. Unlike experts, managers can be moved from node to node so that they are able to function from the most suitable location.



Service names are unique in the meshIQ domain. Contact your meshIQ Platform Administrator for domain naming conventions.

NOTE

Each manager has the following:

- Subordinates called Contacts that are lists of managed experts and managers. Once an expert is included in the contact list, its facts are routed to its Manager for evaluation. Contacts can be experts on any node or on other managers. Managers can be organized into a management hierarchy.
- Managers that subscribe to facts published by their contacts. They make decisions based on Policies and take actions on these facts. Policies are lists of policies that define a manager's behavior based on facts published by the services included in the manager's contact list. (*also see: Policies*).

- Capability of publishing summaries of facts for superior managers (including human managers) and consoles.
- Managers that can be programmed to send notifications via email, pager, or user-defined action.

**IMPORTANT!**

Managers must be deployed on the domain server and/or CEP server before policies can be deployed.

4.6.1 Built-in Managers

There are two built-in Managers.

- Policy Manager
- Speed Manager

4.6.1.1 Policy Manager

The Policy Manager is used to manage policies which are manually deployed within a domain. You will need to deploy at least one policy manager on each CEP server monitored. All policies under this manager must be manually deployed after Startup.

4.6.1.2 Speed Manager

The Speed Manager automatically loads policies from a user-specified speed folder (ds://folder). All business views and business policies located in the speed folder will be loaded automatically upon Manager Startup.

Using the Speed Manager gives users the following advantages over other managers:

- Simplification and ease of use.
- Faster loading time when more than one policy needs to be loaded.
- Easier deployment model.
- Assists in replication time. For example: if there are 10 policies to be loaded, the user only needs to load once. If Policy Manager were used, the user would have to load 10 times (each one separately).

System Services

System Services are an instance of the Speed Manager using the default <server>_facts. All managers load system policies designed to monitor each server.

The default location for System Services can be changed by modifying the following property:

```
property server.system.speed.folder=ds://SYSTEM/nodes
```

Each loaded policy has the same name as the file located under the speed folder with applied manager convention default: `SYS_policy`. The default convention can be changed by defining a property:

```
property server.system.naming.convention=SYS_%name%
```

where `%name%` designates the name of the policy.

4.6.2 Deploying Managers

1. Click **Deployment Tool** to display Directory Viewer.

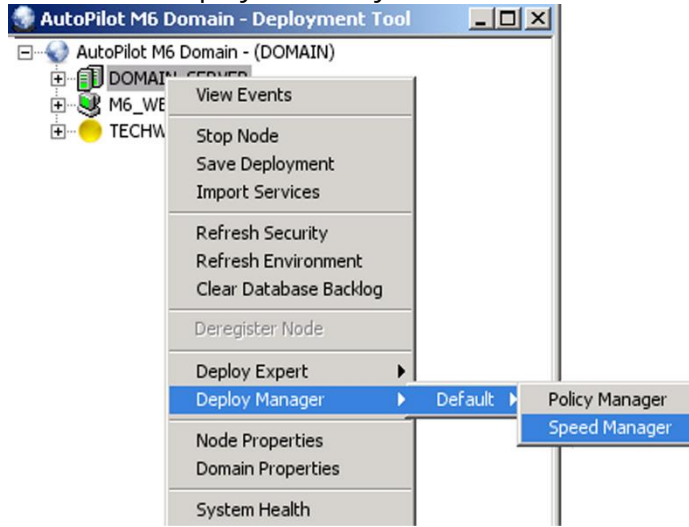


Figure 4-53. Deployment Tool and Networked Nodes

2. Click the node to receive the manager.
3. Right-click the Node to display the Node Menu. Click **Deploy Manager**. The default menu is displayed.
4. Click the Manager. The Manager Properties screen is displayed.

NOTE

1. Each Manager deployed in your meshIQ Network must have a unique name.
2. The configuration and requirements of each Manager are unique. Configuration Screens for each Manager will be different in one or more areas.

5. Review Manager Configuration. (Refer to [Manager Configuration](#) for details.)

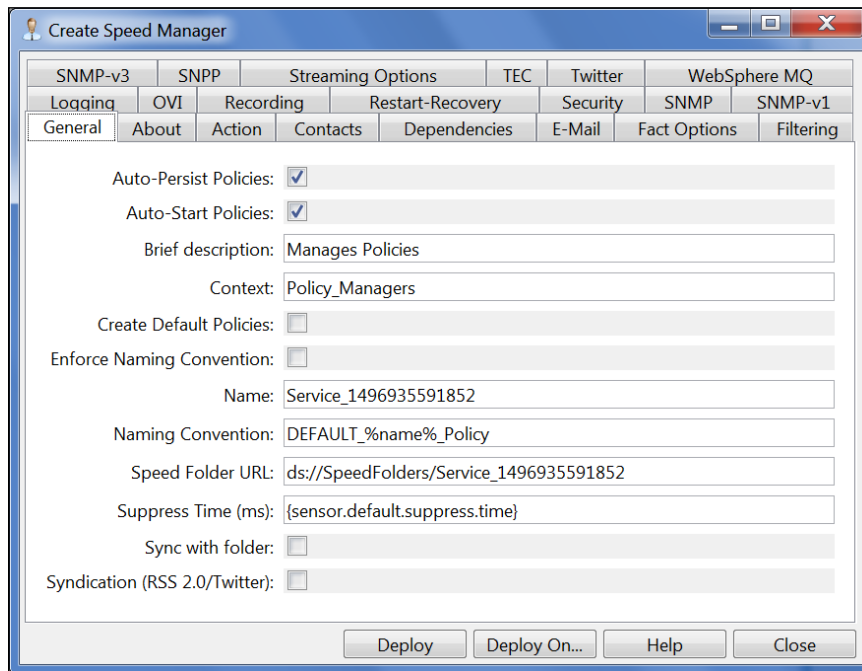


Figure 4-54. Deploying Managers



The **Suppress Time** option allows you to suppress alerts for the time specified. This option is primarily used when testing so the user will not be inundated with alerts.

- Click **Deploy**. The Manager will be deployed on the Node selected in step 2. Click **OK** to verify creation of the new service. The system will create a service name and verify the node you selected for deployment.

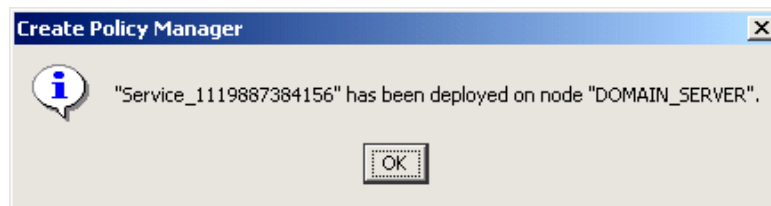


Figure 4-55. Deploy Manager

OR

- Click **Deploy On** to deploy on multiple Nodes or Domains within the meshIQ Network.
- Select *Unique name* format to be used on each Node. Click the preferred naming convention.

Click each node where the manager will be deployed. In *Deploy On* mode all nodes to receive managers must be selected, node selected in step 2 must be re-selected. Keep in mind that each name can only be used once per CEP server/domain server. The check mark in the *Deploy on these nodes* box indicates nodes selected to receive the manager.

- Click **Deploy**.

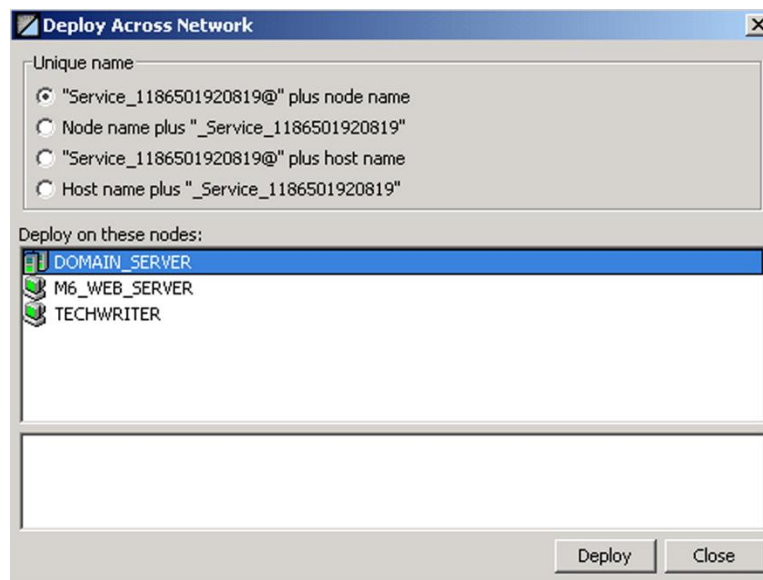


Figure 4-56. Deploy Manager On

4.6.3 Manager Configuration

It is important to fill in "Name" when deploying managers; otherwise, the system will assign a default name that consists of the *Service_{timestamp}* format. Select *Naming Convention* and enable *Enforce Naming Convention* if all policies need to have the same naming convention automatically applied by the manager. Users may provide email and SNPP (Simple Network Paging Protocol) details, when events triggered by deployed policies need to be forwarded via email or pager. Policy alerts can also be syndicated via an RSS news feed. Configure or customize other properties as required.

Table 4-21. Manager Properties

Parameter	Description
General	
Auto-Persist Policies (Speed Manager Only)	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to enable registry persistence for all automatically loaded policies.
Auto Start Policies (Speed Manager Only)	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to auto-start all automatically loaded policies.
Brief description	A short description of the Manager. This name will appear in brackets (e.g.: [Process Monitor]) on screen with listed Node.
Context	User-defined category that will be registered in the Domain Server. Context is displayed as folder icon under each CEP server.
Create Default Policies	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to create default policies that were prepackaged for this manager. Default is disabled.
Enforce Naming Convention	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to enforce naming conventions for all policies. Default is disabled.
Name	The name that uniquely identifies the manager in the domain. Enter or modify Name as applicable, or in accordance with local guidelines. Variations of name are used when deploying services on multiple Nodes. No spaces or blanks are allowed in Service Names.
Naming Convention	Naming convention that applies to all policies. The manager automatically applies the Naming Convention to all subsequent Policies. %name% stands for the name of the policy as supplied by the user.
Speed Folder URL (Speed Manager Only)	Location of all policies loaded automatically upon Manager's Start.
Sync with folder	Synchronize policy deployment with folder contents.
Syndication (RSS 2.0)	Enable/disable policy syndication, via RSS news feed, of policy alerts. By default, all RSS feeds go into <aphome>/rssfeeds. The folder must be created before enabling syndication. You can override the default RSS location by using "property server.rss.folder=<rss_location>/" (must end with a file separator). The feed name will be <manager_name>.xml file.
About	

Table 4-21. Manager Properties	
Parameter	Description
Package Title	Implementation title of the source package.
Package vendor	Name of implementation vendor.
Package version	Version of implementation vendor.
Action	
Enable Action	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to automate actions for all alerts.
User Action	User-defined script or executable triggered for every alert.
Contacts	
Contact List	A space separated list of services managed by this Manager.
Dependencies	
Platform dependencies	Comma-separated list of Operating System platform dependencies.
Service dependencies	Comma-separated list of service dependencies.
E-Mail	
CC Recipients	Comma-separated list of CC recipients (e.g., Bob@gmail.com)
E-mail Notification	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> E-mail notification.
E-mail User Name	User name as defined in the email server (e.g., John)
Enable STARTTLS	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> STARTTLS.
Outgoing Mail Server (SMTP)	Name of SMTP server for outgoing mail.
Primary Recipients	Comma-separated list of primary recipients (e.g.: John.J@email.com)
SMTP Authentication	Select if SMTP server requires authentication.
SMTP Port	SMTP port
SMTP User Name	SMTP user name
SMTP User Password	SMTP user password
Fact Options	
Exclude Expire Filter (regex)	Do not expire facts that match the regexp.
Exclude Fact filters	Comma-separated list of fact paths to exclude during publishing.
Expire Facts(ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size*	Automatically maintains the specified number of samples for each published fact in memory.

Table 4-21. Manager Properties

Parameter	Description
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Fact service alias	Override fact service prefix for all published facts. Facts appear under specified service name.
Include Expire Filter (regexp)	Expire facts that match the regexp.
Include Fact Filters	Comma-separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to Fact History Time or Fact History Size , whichever limit is reached first. If disabled, newer history samples replace older ones on a rolling basis.
Filtering	
Exclude Facts	Filter for excluding facts. For example: MyExpert1*
Include Facts	Filter for including facts. For example: MyExpert2*
Logging	
Audit	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service audit trace.
Log name	Log name associated with the service.
Log service activity	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service activity trace.
Log size (bytes)	Size of Log in bytes. Real log size is the maximum value of server.log.size and logsize.
OVI**	
Application name	Application name reporting the event.
Enable OVI	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> for notifications using Open View Interface (OVI).
Message group	OVI message group for all reported applications.
Object name	Object name associated with reported event.
OVI URL	URL of the HP OVI Event Create Pluglet.
Service name	OVI service name associated with reported event.
Recording	
Anomaly Deviation Limit	Number of standard deviations above or below the mean.
Exclude Filter (regexp)	A regular expression filter to exclude certain facts from being written to the database. Facts have the format expert\class\instance\leaf=value such as in the example Servers\Linux\Serv7\processes=40.
Fact Anomaly Frequency	Frequency at which anomalies are checked and recorded.

Table 4-21. Manager Properties	
Parameter	Description
Fact State Frequency	If Record Fact State is enabled, the value entered here specifies how often the Fact State is updated.
Fact Summary Frequency	If Record Fact Summary is enabled, used to write an intermediate summary record every X th update to the fact during the Summary Interval. This is done to avoid waiting the full Summary Interval for a summary record to appear in the summary table.
Include Filter (regex)	A regular expression filter to include certain facts being written to the database. Same format as described for the exclude filter.
Record Fact Anomalies	Enable/disable fact anomaly recording for this service.
Record Fact History	If enabled, records every fact change into the History database. The exclude/include filters are respected. To define database tables and set options, refer to Recording Facts .
Record Fact State	If enabled, records the last value published (current state) into the state database and restores that value when the CEP Server is stopped and restarted. The exclude/include filters are respected. To define database tables and set options, refer to Recording Facts .
Record Fact Summary	If enabled, records summary record at the interval designated in the Summary Interval (ms) field into the Summary database. The exclude/include filters are respected. To define database tables and set options, refer to Recording Facts .
Storage for Anomalies	SQL table where anomalies are recorded.
Storage for History	Database table where the Fact History data is stored.
Storage for State	Database table where the Fact State data is stored.
Storage for Summary	Database table where the Fact Summary data is stored.
Summary Interval (ms)	If Record Fact Summary is enabled, designates how often the Fact Summary data is written (in milliseconds).
Restart-Recovery	
Automatic Start	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> automatic start.
Save in Registry	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> persistent services saved in Registry.
Synchronous Control	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> synchronous service initiation.
Security	
Inherit permissions from owner	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to allow inheriting of permissions from the owner's permission mask.
Owner	User that owns this object.
Permissions	Permission for users in the same user group and others, see Account Permission Masks in Service and Account Permissions for details.
SNMP	

Table 4-21. Manager Properties

Parameter	Description
Debug level (0-15)	SNMP debug level; 0 is the lowest level.
Enable SNMP-v1	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> notifications using SNMP-v1.
Enable SNMP-v2	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> notifications using SNMP-v2.
Enable SNMP-v3	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> notifications using SNMP-v1.
SNMP v2/v3 trap oid	Trap Object Identifier (OID): 1.3.6.1.6.3.1.1.4.1.0 which identifies the notification being sent as a trap.
Trap hostname	Name of the host where traps are sent.
Trap port	Port number of the trap listener.
User variable OID	Object identifier of a user-defined variable to be added to every trap, perhaps to identify the generator of the trap. Consult the SNMP system administrator for a valid OID that can be used. It could be a pre-existing SNMP variable, such as sysName, whose OID is 1.3.6.1.2.1.1.1.
User variable value	Text string value of the user variable. For example: alpha.meshIQ.com.
SNMP-v1	
Enterprise OID	SNMP enterprise OID. The dotted numeric path, which always starts with 1.3.6.1.4.1 (iso.org.dod.internet.private.enterprises) and ends with the object ID assigned by Internet Assigned Numbers Authority (IANA) to the vendor of the device being monitored. For example: If using meshIQ's object ID of 1796, then the SNMP Enterprise OID would be 1.3.6.1.4.1.1796, which is used in the context of meshIQ's WMQ MIB. You must know the enterprise ID for the vendor of the device or software being monitored. For example: Cisco, Juniper Networks, or meshIQ.
Generic trap	Generic trap indicator: 0 - coldStart 1 - warmStart 2 - linkDown 3 - linkUp 4 - authenticationFailure 5 - egpNeighborLoss 6 - enterpriseSpecific If 'Generic Trap 6' is specified, it is always followed by 'Specific Trap n'.
Specific trap	A number identifying the specific trap number.
Trap community	Name of community (group) receiving traps. This is a security feature in SNMP-1 to prevent unauthorized monitoring of your devices or generating spoofed traps.
SNMP-v3	
Authentication Protocol	Encryption protocol used during password authentication.

Table 4-21. Manager Properties

Parameter	Description
Authentication	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> authentication.
Context engine id	SNMP context engine Identification.
Context name	SNMP context name.
Privacy password	Privacy password.
Privacy	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> privacy password.
User name	Authentication user name.
User password	Authentication user password.
SNPP	
Enable Paging	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> SNPP paging.
Outgoing Paging Server (SNPP)	Host name where SNPP is running.
Pager PIN	Pager's Personal Identification Number.
Paging Server Port	Port where SNPP server is listening. (See System Administrator for additional assistance).
Streaming Options (Refer to Streaming Data.)	
Application name	Sets application name
Data center name	Sets data center name
Exclude filter (regexp)	Ignore facts that match specified regular expression
Include filter (regexp)	Log facts that match specified regular expression
Location	Sets server location
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework)
Streaming configuration	Streaming configuration block name
TEC***	
Enable Tivoli TEC	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> notifications using Tivoli Enterprise Console (TEC).
Event Attributes	Assigns a value to any valid attribute. The attribute should be one defined for the event class. Separate multiple attribute=value expressions with spaces.
Event class	Specifies class of the event. It must match a class that is configured at the server.
Event message	Text of the event in double quotation marks.
Event source	Specifies the source of the event. If any blank spaces are in the source name, enclose the source name in double quotation marks.
TEC command	Path to the command that sends events to a TEC event server.

Table 4-21. Manager Properties

Parameter	Description
TEC config file	Specifies the name of TEC/wpostzmsg configuration file.
TEC server	Specifies the name of event server in name registry format. Enter @EventServer to have events submitted to locally named event server.
Twitter	
Twitter	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> Twitter posting of policy events.
Twitter Access Key	Twitter OAuth access key.
Twitter Access Secret	Twitter OAuth access secret.
Twitter Password (deprecated)	Twitter user account password.
Twitter User	Twitter user account name.
WebSphere MQ (IBM MQ)	
Facts to WebSphere MQ	Enable/disable writing facts to IBM MQ
Password	IBM MQ user password
Queue manager host	Name of the host where queue manager is defined
Queue manager name	Name of local queue manager
Queue manager port	Channel listener TCP port for the queue manager
Queue name	Name of queue where facts will be sent
Server connection channel	Name of the SRV CONN channel on the queue manager
SSL certificate store	SSL certificate store location
SSL cipher suite	Cipher suite that matches the CipherSpec of the channel
SSL enable	Enable SSL communication for channel communication
User ID	IBM MQ user ID

* **Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to [Fact Volatility](#) for more detailed information on fact volatility.)

- ** If **OVI** selected, users must install and configure OVI and make oviEventCreatePluglet available. Health severities are mapped to OVO severities as follows:

Table 4-22. Core Services and OVO Health Severities	
Core Services Severity	OVO Severity
UNKNOWN	UNKNOWN
EMERGENCY	CRITICAL
CRITICAL	CRITICAL
FAILURE	MAJOR
ERROR	MINOR
WARNING	WARNING
SUCCESS	NORMAL
DEBUG	UNCHANGED
INFO	NORMAL
NOTICE	NORMAL

*** If **TEC** selected, Core Services health severities are mapped to TEC as follows:

Table 4-23. Core Services and TEC Health Severities	
Core Services Severity	TEC Severity
UNKNOWN	UNKNOWN
EMERGENCY	FATAL
CRITICAL	CRITICAL
FAILURE	CRITICAL
ERROR	MINOR
WARNING	WARNING
SUCCESS	HARMLESS

4.7 Policies

Policies reside within one or more managers. They are triggered upon changes in facts or upon any other built-in conditions supplied by users using business views wizard. These policies can vary in complexity depending on the automation tasks they are assigned to accomplish. A policy that must take into account a number of different conditions before the manager can act is more complex. Complex policies perform data correlation and analysis.

**NOTE**

1. The configuration of each policy is unique and depends on the user-defined rules, thresholds, and criteria. Policies can only be deployed within managers. Deploy managers first if required.
2. The names of policies must be unique within a single manager.

This section covers the deployment and configuration of policies. The policies outlined in this section are four-level templates. They define non-application specific Policy development. Policies designed for specific applications are addressed in the supporting documentation for that software or option. The following policy template configurations are discussed in this section:

- **New View:** Loads specified user-defined business views into a manager. Business views are created using the User Console. Once loaded, business views run in the background and act on behalf of the owner.
- **New Filter:** Contains all the attributes of the view policy and is used by managers to filter information from Manager's contacts. The information is then published on behalf of the Manager that hosts the Filter Policy.
- **New Schedule:** Contains all the attributes of the filter policy and is a schedule policy that can schedule start-up/shutdown of any service/experts/managers including policies. Services can be initiated at any time for a specified duration (in minutes) and any day of the week.
- **New Persistence:** Contains all the attributes of the schedule policy, plus maps facts to any database table. It allows users visually map facts to DB table and save the mapping schema to a schema file. These schema files are saved with `.TBL` extension. The schema can then be deployed using persistence policy within any manager.

**TIP**

Right-click the target manager and select "Deploy Policy" menu option, to deploy policies. Business views should be deployed using Business View Tool or Business View Explorer.

4.7.1 Deploying Policies


IMPORTANT!

Managers must be deployed on the domain server and/or CEP server before policies can be deployed.

1. Click **Deployment Tool** to display Directory Viewer.


NOTE

Policy icons are assigned by the system and are random. There is no functional deference implied.

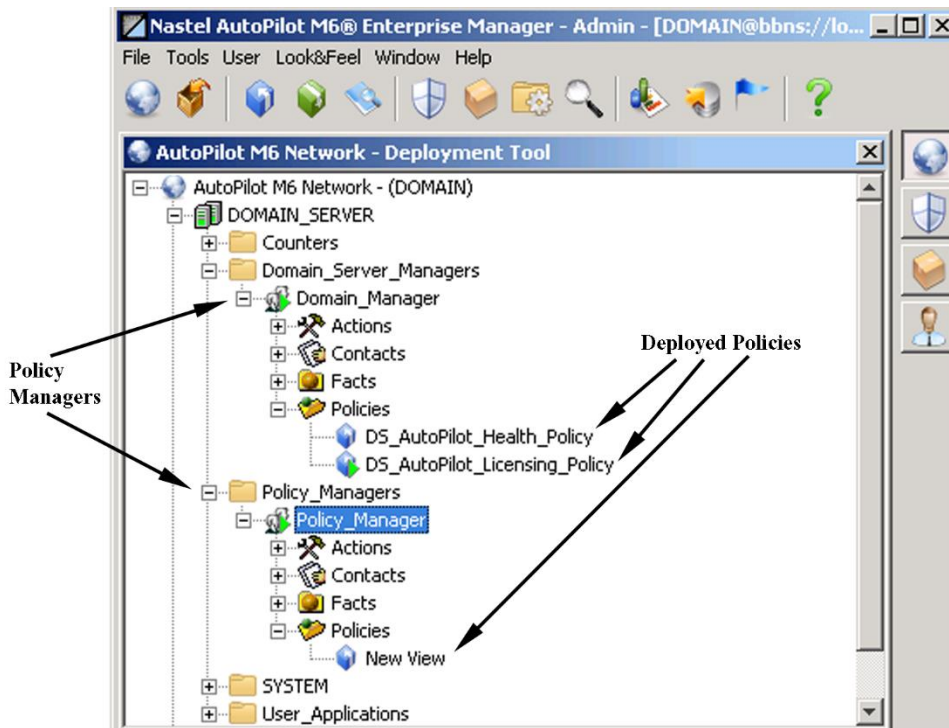


Figure 4-57. Deployed Policies

2. Right-click the manager to receive the policy to display Menu. Click **Deploy Policy**. The Policy sub-menu is displayed.
3. Click desired Policy type to be deployed from the groups displayed.


NOTE

Built-in Policies may vary based on installation options exercised and local meshIQ Platform deployment requirements. The default Policy templates are provided to enable users to develop custom Policies for local use.

4. Select the Policy to be deployed from the sub-menus. The configuration screen for the selected Policy will be displayed.

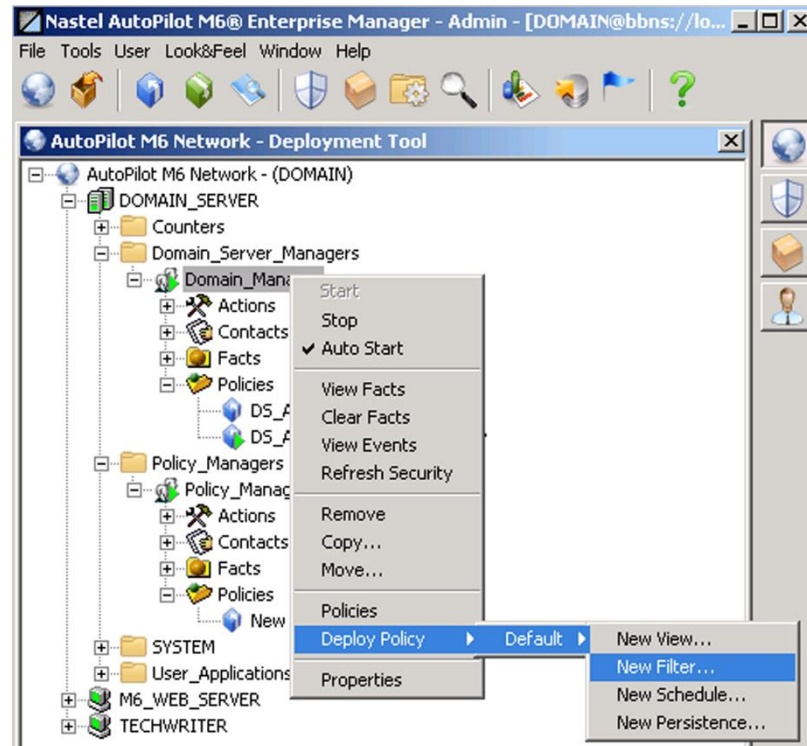


Figure 4-58. Policy Deployment



Configuration and requirements are unique for each Policy. Configuration Screens for each Policy will be different in one or more areas.

5. Review Policy Configuration. Refer to [Configuring Policies](#) for details of policy parameter configuration.
6. Click **Deploy**. The Policy will be deployed with the Manager selected in step 2.

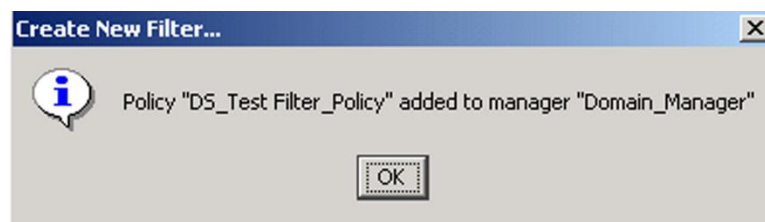


Figure 4-59. Policy Created

OR

7. Click **Deploy On** to deploy on multiple managers within domain or node. Click the manager to receive the policy and click **Deploy**. Deploy the policy within as many managers as needed; each time you deploy the policy an acknowledgement statement is displayed. The green check mark indicates successful deployment.

8. Click **Close** once you have deployed the policy where required.

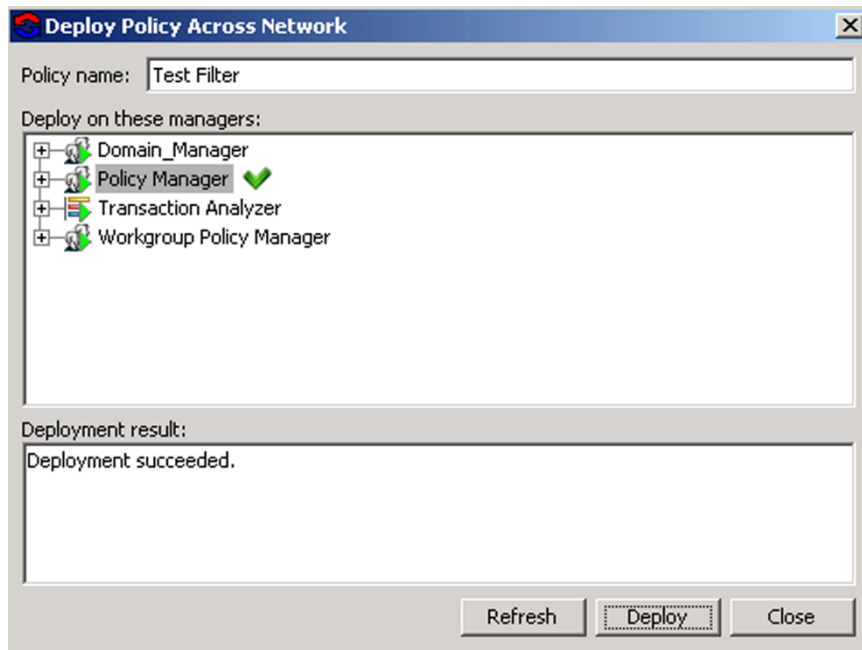


Figure 4-60. Policy Selection and Deployment

Deployment View

To display all deployed policies, click **Tools > Deployment View** from the toolbar. The deployment panel displays server name along with manager name (.manager@server label).

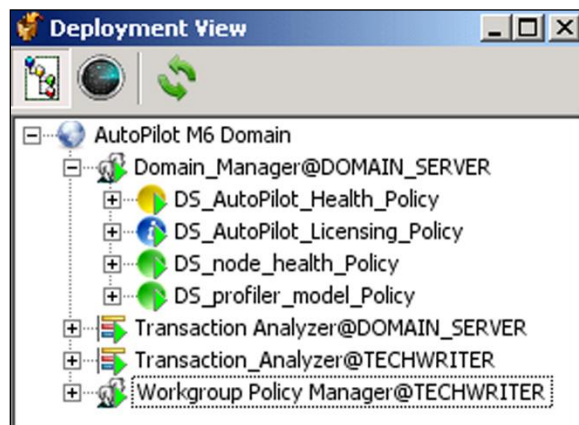


Figure 4-61. Deployment View

Sensor Search

There is an option to search within policies to find a specific sensor without having to expand all nodes.

1. Right-click a policy and select **Open** from the popup menu.

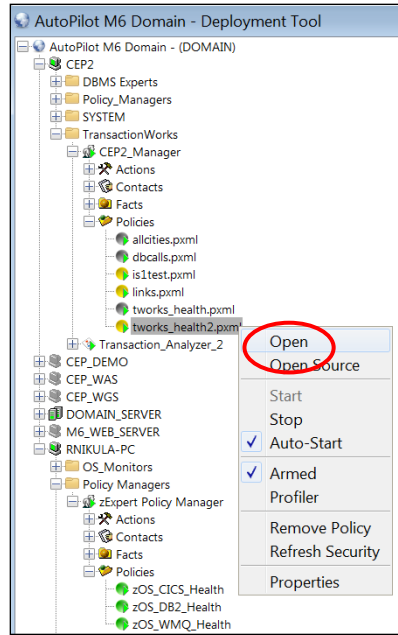



Figure 4-61A. Policy Menu

2. Click the **Filter Sensors** icon  to display the sensor list which allows you to search.

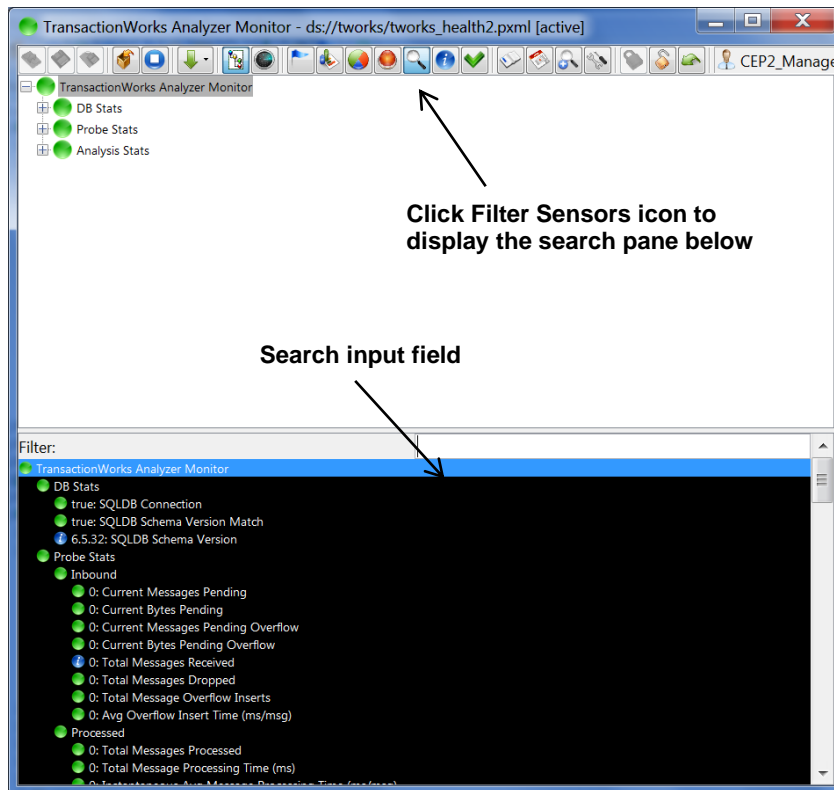


Figure 4-61B. Sensor Search List

- In this example, we searched for a sensor with **lag** in its name. The result was the **Analysis Time Lag (ms)** sensor. Click **Analysis Time Lag (ms)** to expand the tree in the top pane to see where the sensor is located (Figure 4-61D).

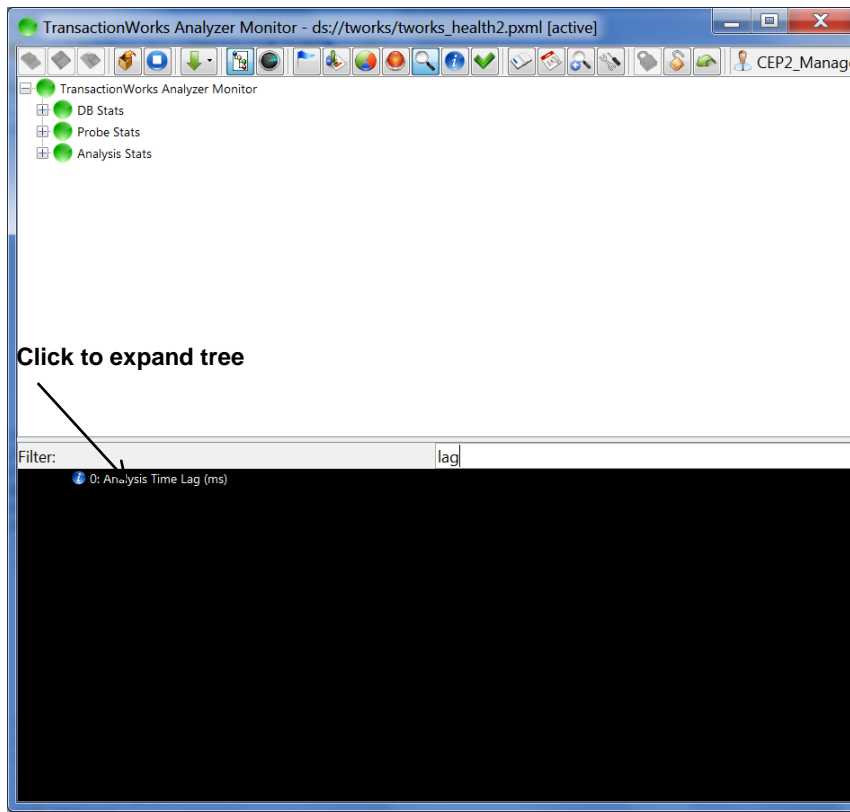


Figure 4-61C. Filtered Sensor List

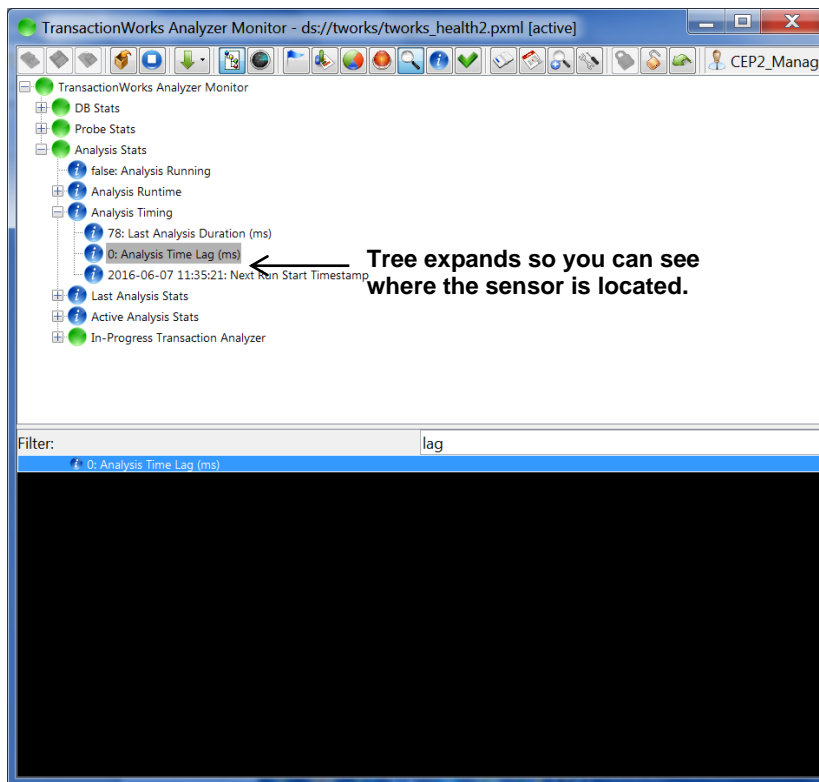


Figure 4-61D. Expanded Tree

4.7.2 Configuring Policies

The built-in policy properties are identified below. Properties that are unique to a policy are annotated accordingly. All other properties are common to all policies. Configure or customize the policy as needed for your applications.



When making changes to a policy, the changes are logged to a file in the Domain Server. These audit records are viewed by right-clicking the Domain Server and selecting View Events from the popup menu.

Table 4-24. Policy Properties

Parameter	Description
General	
Brief Description	A short description of the policy. This name will appear in brackets (for example: [Process Monitor]) on screen with listed Node.
Name	The default name is system assigned. Enter or modify name as applicable or in accordance with local guidelines. Variations of defined names are used when deploying Policies on multiple Managers/Nodes.
Data Base Scheme File (Persistence Policy only)	Data base scheme file determined by the database tool (example: /opt/meshiq/mytable.tbl)
Schedule Services (Schedule Policy only)	Comma-separated list of services to schedule (for example: Expert1, Manager, Policy@manager1)
Armed (New View Policy only)	Disarms/Arms the policy. Action notifications are enabled when Policy is armed.
Business view file (New View Policy only)	File name created by Business View Tool (for example: ds://AutoPilot_EE/myfile.bsv)
Publish stats (New View Policy only)	Disables/enables publishing of Policy statistics.
Publish view (New View Policy only)	Disables/enables publishing of Policy view from web client.
About	
Package Title	Implementation title of the source package.
Package Vendor	Name of implementation vendor.
Package Version	Package version as assigned by the vendor.
Dependencies	
Platform dependencies	Comma-separated list of Operating System platform dependencies.
Service dependencies	Comma-separated list of service dependencies.
Fact Options	
Exclude Fact Filters	Comma-separated list of fact paths to exclude during publishing.

Table 4-24. Policy Properties

Parameter	Description
Expire facts(ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size*	Automatically maintains the specified number of samples for each published fact in memory.
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Include Fact Filters	Comma-separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to Fact History Time or Fact History Size , whichever limit is reached first. If disabled, newer history samples replace older ones on a rolling basis.
Filtering	
Filter Facts	Comma-separated list of fact filters.
Logging	
Audit	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service activity trace.
Debug Trace	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> business view debug trace mode.
Log Name	Log name associated with the service.
Log Service Activity	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> service activity trace.
Log size (bytes)	Log size in bytes. Real logsize is the maximum value of server.log.size and logsize.
Performance	
Index Relations	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to control the indexing of sensors. This allows inter-policy cross referencing. It is not recommended to use this option when speed and CPU usage is an issue.
Suppress Duplicate Value Propagation	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to suppress propagation of duplicate sensor values to parent sensors. <i>Recommended configuration</i> . Disabling will cause memory and CPU degradation.
Suppress Real-time Health Aggregation	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to suppress real-time sensor health aggregation. <i>Recommended configuration</i> . Disabling will cause memory and CPU degradation.
Profiler	
Profile sensor performance	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> profile sensor performance for each sensor. Default is disabled.
Profile state counts	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> profile state counts for each sensor. Default disabled.
Profile state timing	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> profile state timing for each sensor. Default disabled.

Table 4-24. Policy Properties

Parameter	Description
Profile thread	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> profile sensor worker thread performance. Default enabled.
Profile sampling rate(ms)	Profiler sampling and publishing rate in milliseconds. Default 30000.
Sensor profiler	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> sensor profiling. Default is disabled.
Restart and Recovery	
Automatic Start	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> automatic start.
Save in Registry	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> persistent services saved in registry.
Synchronous Control	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> synchronous service initiation.
Security	
Inherit permissions from owner	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> to allow inheriting of permissions from the owner's permission mask.
Owner	User that owns this object.
Permissions	Permission for users in the same user group and others, see Account Permission Masks in Service and Account Permissions for details.
Database (New Persistence Policy Only)	
Data source name (DSN)	Logical data source name that points to the physical table.
Database Table	Name of the physical Database table (Example: CPU_Table).
DB logon ID	Logon ID required to access database.
DB Password	Password to access the database.
JDBC driver	Class name of the JDBC Driver provider.
Schedule (New Schedule Policy Only)	
Begin Date	Date to trigger service (example: 09/20/2024).
Begin Time	Time to trigger service (example: 12:00 PM EST).
Duration (mins)	Maximum time allocated for each service (in minutes).
Expiration Date	Date to end services (example: 06/30/2025).
R-R Service Timer (ms)	Time in (ms) given for each service to complete (example: 60000 ms = one minute).
Round Robin Delay (ms)	Delay (ms) before next service is initiated by the schedule (R-R only).
Round Robin Schedule	Round Robin schedules services one after another.
Schedule Every	Comma-separated list of days scheduled. (example: Day (every day), Monday, Tuesday, etc.)

* **Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact

history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to [Fact Volatility](#) for more detailed information on fact volatility.)

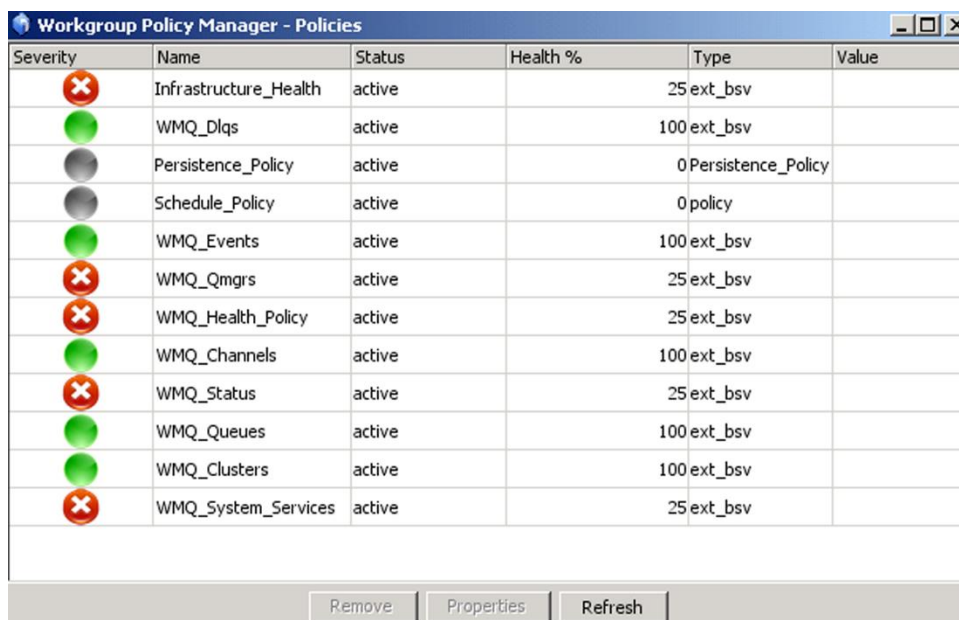
To control policy execution and performance, the following properties are used:

- **server.sensor.delivery.flowpct=0.** Setting the value between 0 and 100 temporarily suspends the flow to business views when the usage reaches 100%. The flow will remain suspended until the usage drops below the value specified by the variable. Setting the value to 0 or below, disables this property and the flow will always stay enabled. Enabling the flow control puts a maximum limit on the number of outstanding events for each business view and limits the memory growth associated with unlimited growth of queued events.
- **server.sensor.delivery.batch=2000.** The processing of sensor events is done in batches. It is not recommended to set this value below 100. It may create too much thread contention and decrease performance of the rule engine.

4.7.3 Using Managers for Displaying Policy Information

Information about all policies under a manager can be displayed by right-clicking on the manager and selecting Policies. This information includes:

- Name of policy
- Severity of policy in icon format
- Status – active, unknown, stopped, etc.
- Health percentage – 0% to 100%
- Type – Business View (bsv), policy, business process (bsp), etc.
- Value – if previously defined.



Severity	Name	Status	Health %	Type	Value
✖	Infrastructure_Health	active		25 ext_bsv	
●	WMQ_Dlqs	active		100 ext_bsv	
●	Persistence_Policy	active		0 Persistence_Policy	
●	Schedule_Policy	active		0 policy	
●	WMQ_Events	active		100 ext_bsv	
✖	WMQ_Qmgrs	active		25 ext_bsv	
✖	WMQ_Health_Policy	active		25 ext_bsv	
●	WMQ_Channels	active		100 ext_bsv	
✖	WMQ_Status	active		25 ext_bsv	
●	WMQ_Queues	active		100 ext_bsv	
●	WMQ_Clusters	active		100 ext_bsv	
✖	WMQ_System_Services	active		25 ext_bsv	


Remove Properties Refresh

Figure 4-62. Policy Information

Selecting a policy enables **Remove** and **Properties** buttons. The policy list can be refreshed at any time by clicking **Refresh**.

Double clicking a bsv opens it.

Double clicking a policy displays its property screens.


NOTE You can also view this information by right-clicking **Policies > Manage**.

4.7.4 Policy Profiler

The performance of each business view can be tracked by enabling its profiler. Once enabled, the profiler metrics are published under individual managers in the *Facts\Profiler* subfolder. To enable the policy profiler, right-click the policy name and select *Profiler* (see Figure below) or enable *Sensor Profiler* under the *Profiler* tab in Properties. (Refer to [Table 4-24](#).) A check mark indicates *Enabled*.

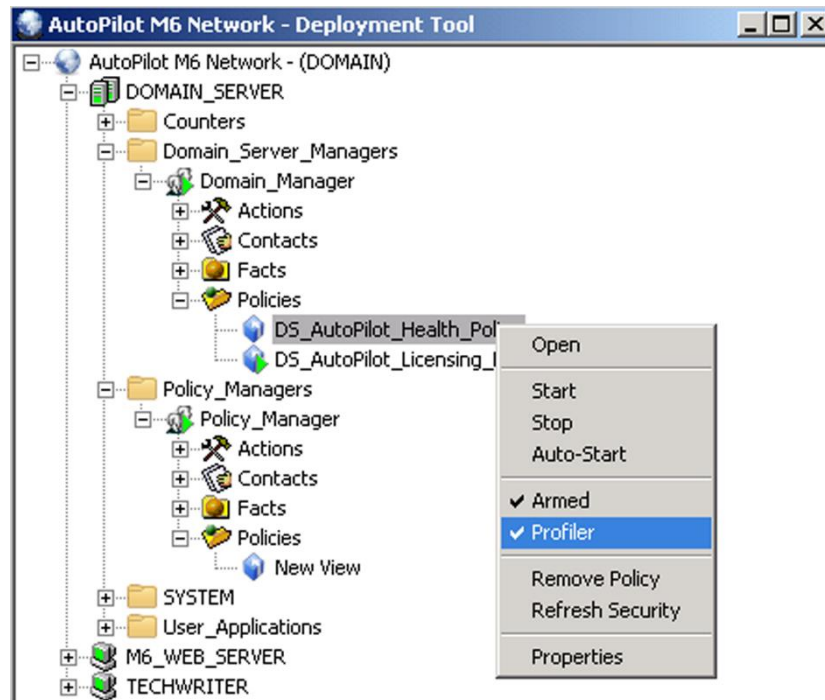


Figure 4-63. Enabling Policy Profiler

Profiler metrics are shown below.

Table 4-25. Profiler Metrics Per Policy	
Root Sensor Only	Thread\sensor_delivery_backlog
	Thread\sensor_rule_backlog
	Thread\sensor_delivery_rate_per_sec
	Thread\sensor_arrival_rate_per_sec
	Thread\sensor_worker_thread_alive

Table 4-25. Profiler Metrics Per Policy	
	Thread\sensor_total_delivered_events
	Thread\sensor_total_arrived_events
	Thread\sensor_total_dropped_events
	Thread\sensor_delivery_limit
	Thread\sensor_worker_thread_full
	Thread\sensor_worker_thread_util
	Thread\sensor_worker_thread_flow

Table 4-26. Profiler Metrics Per Sensor	
Action Related Metrics	Performance\Actions\sensor_last_action_rc
	Performance\Actions\sensor_last_action_exec_time_ms
	Performance\Actions\sensor_action_failures
	Performance\Actions\sensor_action_execs
	Performance\Actions\sensor_action_timeouts
Rule Performance Related Metrics	Performance\Rules\sensor_rule_exec_count
	Performance\Rules\sensor_last_exec_time_ms
	Performance\Rules\sensor_peak_exec_time_ms
	Performance\Rules\sensor_min_exec_time_ms
	Performance\Rules\sensor_rule_instr_count
	Performance\Rules\sensor_rule_rate_per_sec
	Performance\Rules\sensor_rate_per_sec
	Performance\Rules\sensor_idle_pct
	Performance\Rules\sensor_waiting_pct
General Run-Time	Performance\General\sensor_current_state
	Performance\General\sensor_runtime_ms
	Performance\General\sensor_time_waiting_ms
	Performance\General\sensor_child_count
	Performance\General\sensor_uptime_sec
Obtain State Counts – Number of times sensor spent in each state	StateCounts\sensor_idle_count
	StateCounts\sensor_running_count
	StateCounts\sensor_stopped_count
	StateCounts\sensor_starting_count
	StateCounts\sensor_stopping_count
	StateCounts\sensor_action_count
	StateCounts\sensor_subscribing_count
	StateCounts\sensor_error_count

Table 4-26. Profiler Metrics Per Sensor	
	StateCounts\sensor_requesting_count
	StateCounts\sensor_waiting_count
Time Spent in Each State	StateTiming\sensor_idle_ms
	StateTiming\sensor_running_ms
	StateTiming\sensor_stopped_ms
	StateTiming\sensor_starting_ms
	StateTiming\sensor_stopping_ms
	StateTiming\sensor_action_ms
	StateTiming\sensor_subscribing_ms
	StateTiming\sensor_error_ms
	StateTiming\sensor_requesting_ms
	StateTiming\sensor_waiting_ms

4.7.5 Health and Load Balancing Policy

The SYS_node_health.bsv monitors a hosting CEP server’s health and automatically adjusts its load. It retains 1.0MB worth of historical data about node key performance indicators in server_health.log under \naming and \localhost for each running instance. The data is helpful during problem determination. This user-defined policy is automatically installed on every node by the [node]_Facts system service. Users can modify it to meet their specific needs. By default, the policy source is located in ds://SYSTEM//nodes folder.

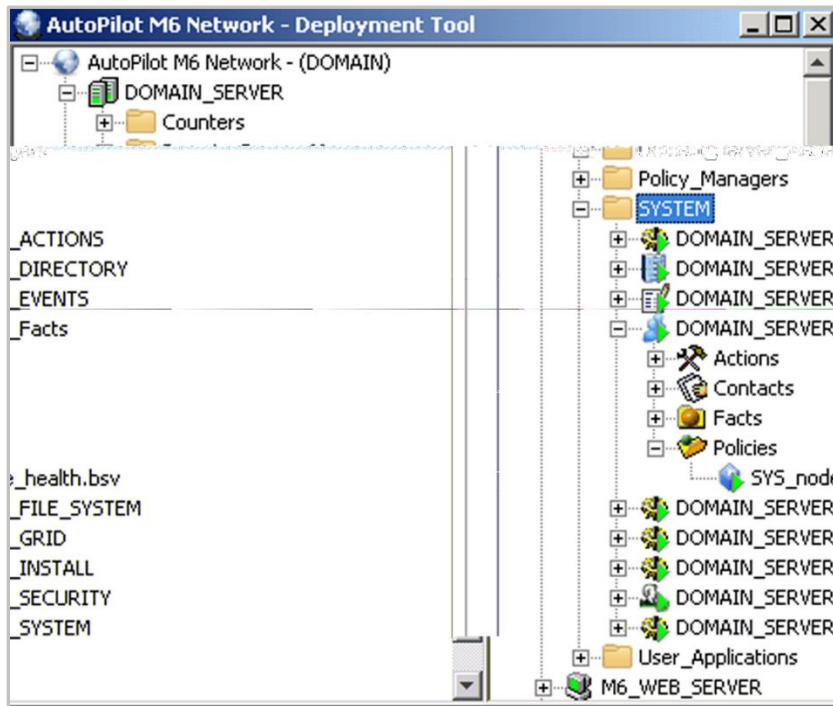


Figure 4-64. System Service

The Response Time Delay drains an activity if the response time is over a user-defined threshold and automatically stops all user-defined services when a threshold is reached. The policy response time rate is in milliseconds.

The Service Status and Recovery monitors the status of all local services registered in the domain and starts them automatically, if Auto-Start is set, when it is no longer overloaded.

Both sensors work together to load balance the hosting CEP server.

When defining properties, the *User Action* field under the *Alert* tab is automatically populated for Service Restart and Recovery and Response Time Delay. The properties *Sensor\sensor_turn_around_time_ms* and *Topic\fact_delivery_turn_around_ms* are used to measure how much time it takes for facts to be processed and displayed by the policies. Alerts can be set up to monitor these metrics and alert the user prior to a load unbalance. The sum of both metrics is the total delay incurred by the system due to load. The longer the time, the less responsive the CEP server becomes. The user is alerted when the sum exceeds a user-defined limit which is usually 30000 ms.

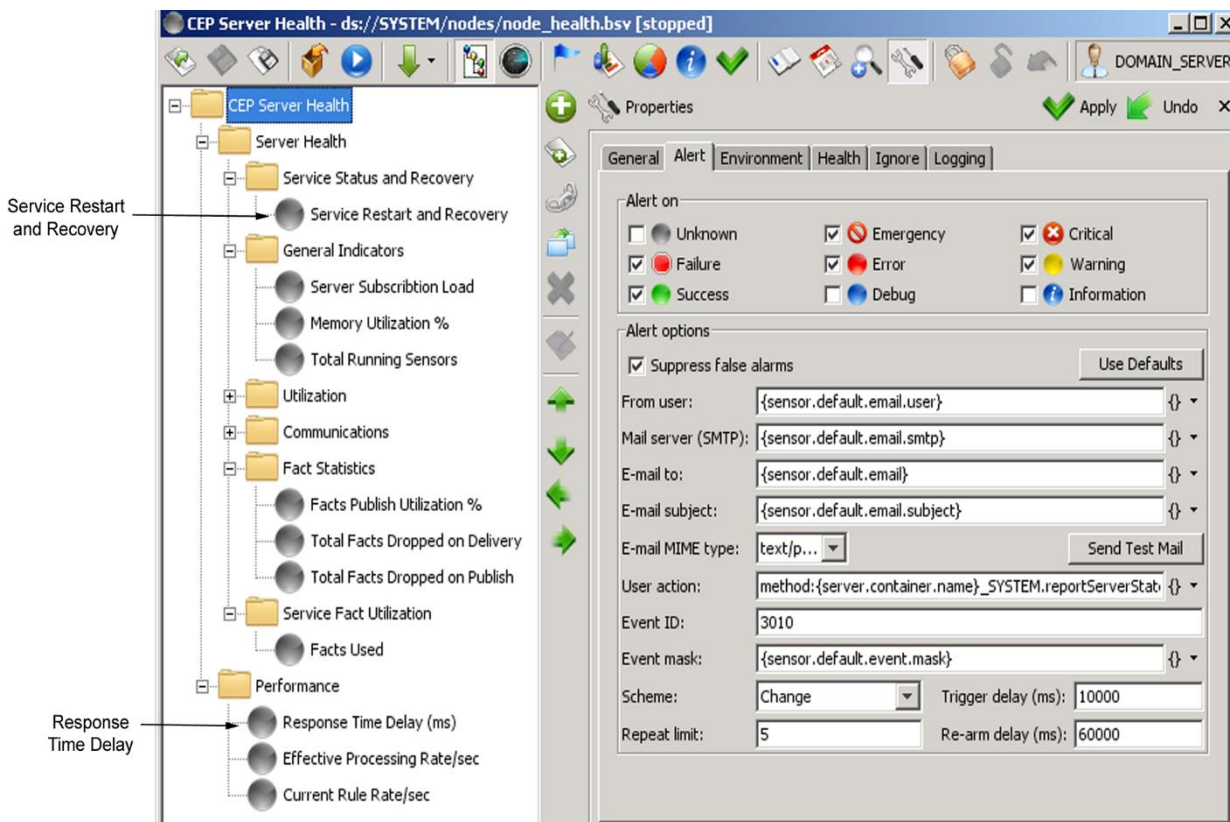


Figure 4-65. Default Alert Tab Properties Screen

The *User action* field can be cleared by highlighting the entire field and deleting it using your keyboard.

The Auto Stop/Start action can be enabled by clicking on the down arrow to the right of the *Scheme* field and selecting *Change*.

The environment variables used in the CEP server Health.bsv include the following:

MID_LIMIT=8000

HIGH_LIMIT=25000

If the top-level CEP Server Health sensor folder is selected and **Properties** > **Environment** tab is selected, the two environment variables from above are visible:

MID_LIMIT=8000

HIGH_LIMIT=25000

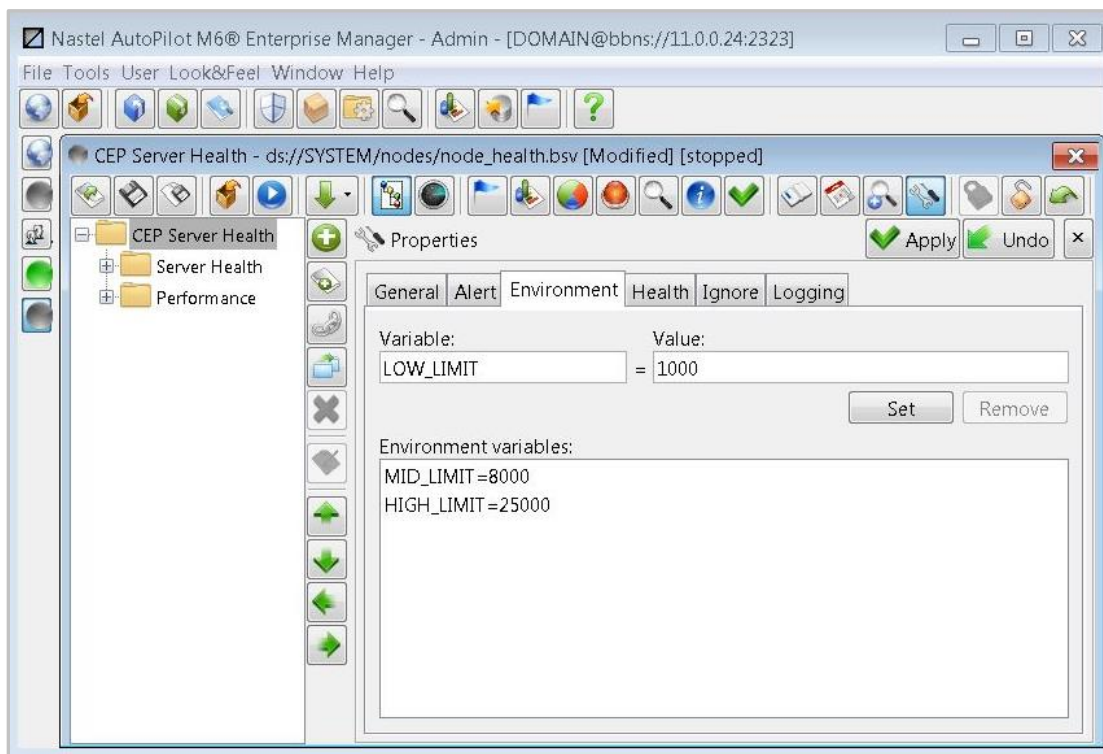


Figure 4-65-A. Adding a User-defined Environment Variable to a Policy

These can be used in any sensor of the policy. The figure above shows the two environment variables in the Environment tab of the policy properties.

Detail steps:

1. Select the policy SYS_node_health.bsv under DOMAIN_SERVER > SYSTEM > DOMAIN_SERVER_FACTS > Policies.
2. Right-click and select Open.
3. Right-click on the top-level folder with policy name CEP Server Health, select Properties.
4. Select the Environment tab.
5. An additional environment variable LOW_LIMIT can be added, as shown in the figure, by entering the Variable Name and Value and clicking Set.

- Then click the **Save** icon on the toolbar (2nd from left). When the confirmation dialog to save the changes appears, click **yes**.

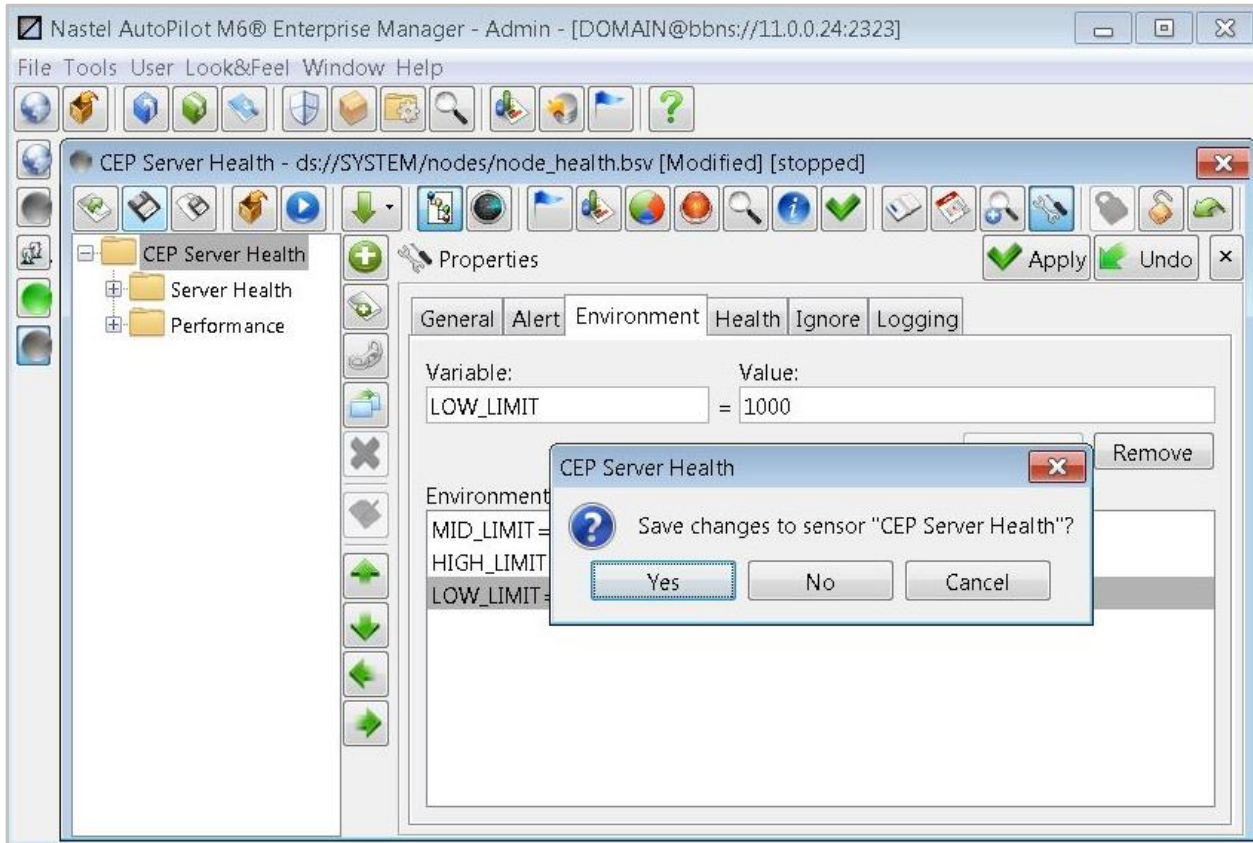


Figure 4-65-B. Adding a User-defined Environment Variable – Save Changes

The three environment variables can all be used in any sensor of the CEP Server Health policy.

The related business view (BSV) file is updated with the new env var inserted with the other two:

naming\policies\SYSTEM\nodes\node_health.bsv:

```
<Value>MID_LIMIT=8000,HIGH_LIMIT=25000,LOW_LIMIT=1000</Value>
```

If you want to define an environment variable for use by any one sensor, select the sensor and add the environment variable similarly as done above for the entire policy.

To use the env var in a sensor, right-click a sensor, such as Service Restart and Recovery, select Wizard, and click through to the Dynamic Sensor Options, where the If-Then-Else rules are located.

Sensor Wizard - Service Restart and Recovery

Dynamic Sensor Options

Dynamic sensor name

Use fact name [%f0] {server.container.name}_SYSTEM\Services*\Status

Use part of fact name [%f0] {server.container.name}_SYSTEM\Services*\Status

User defined name Service %2

Dynamic sensor options

Summarize: Expire dynamic sensor seconds after sensor health >= 1.0

Define rules for dynamic sensors: Environment Function + - * / % () Action

+ X Severity: Emergency Actions: Show value: %f0:Value

+ X (...) When %f0:Value is not active

+ X (...) And %f1:Value is true

+ X (...) And %f2:Value + %f3:Value + %f4:Value is at least {HIGH_LIMIT}

+ X Severity: Critical Actions: Show value: %f0:Value

+ X (...) When %f0:Value is not active

+ X (...) And %f1:Value is true

+ X (...) And %f2:Value + %f3:Value + %f4:Value is greater than {MID_LIMIT}

+ X Severity: Warning Actions: Show value: %f0:Value

+ X (...) When %f0:Value is not active

+ X (...) And %f1:Value is true

+ X (...) And %f2:Value + %f3:Value + %f4:Value is at most {MID_LIMIT}

< Back Next > Start Over Cancel

Figure 4-65-C. Policy CEP Server Health Sensor Wizard Service Restart and Recovery

Note how HIGH_LIMIT and MED_LIMIT are used.

If you select one of those fields, the env var choices will appear and the **Environment** button on the **Define rules for dynamic sensors** line is activated, to indicate you can make an env var choice.

4.8 Business Views

Business view is a collection of rules that define the desired state of the environment. By identifying and correlating facts, the user can graphically represent the state of the environment's applications and middleware, and the impact on each in the event of any degradation or failure.

Business views are customized to present the information in the way best suited for the user. Business view offers notifications by *email* and/or pager. Actions can be associated to any business view to create powerful automation for any application.

Business views can be accessed or shared via the User Console. Refer to [Appendix D](#) for Best Practices.



New business views are saved with the extension .pxml to support standard java XML serialization. Support for the .bsv extension used in previous versions still exists.

4.8.1 Business View Deployment Cycle

Business views cannot be started unless deployed. Please review the deployment cycle below. All business views must be deployed before they can be activated.

Business View deployment cycle goes through the following stages:

- **Develop** – user or a team creates a business view, which is usually stored on the local file system.
- **Check-in** – business views must be saved or uploaded into central domain repository (using Business View Explorer or Business View Tool).
- **Deploy** – business view is assigned to run on one or more managers. At this stage all user-defined rules, alerts and automations are being delegated to the selected manager(s). All the resources such as user actions and scripts, database definitions, and file definitions must exist on the server where the assignment occurs.



When starting the business views, the Business View Tool will automatically trigger the deployment cycle. The deployment cycle must be successfully completed before a business view can be started. Make sure all business views are saved onto a Domain Server.

4.8.2 Exploring and Managing Business Views

Business View Explorer is a tool supplied with the console and can be launched from the console's *Tool* menu. Use Business View Explorer to:

- Explore business view domain repository
- Save/Export sensor information such as facts, logic, conditions, and alerts to an HTML, PDF, RTF, or XML file.
- Deploy business views from the repository onto one or more managers
- View where business views are deployed
- Set permissions for ownership and access by users and groups
- Lock and check out business views/check in and unlock business views.
- Upload locally created business views to domain repository
- Download business views from the domain repository to your local system

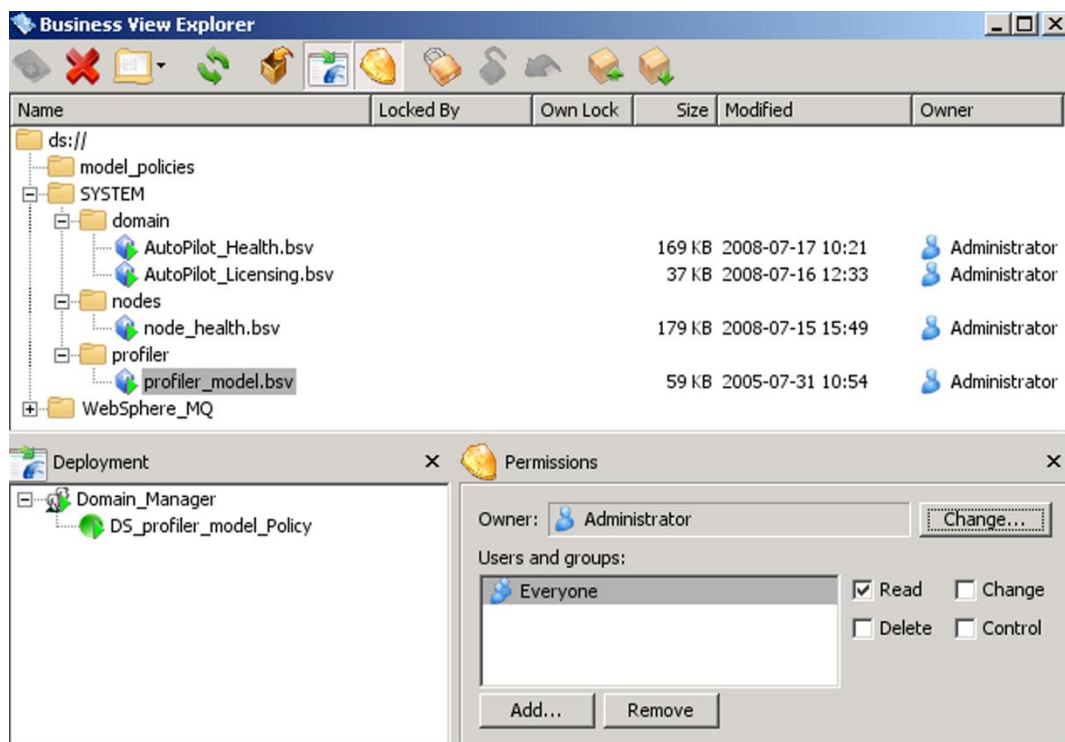


Figure 4-66. Business View Explorer


4.8.2.1 Setting Business View Permissions and Access

Permissions are set to specify ownership and access control to prevent unauthorized manipulation of a business view. Each business view has a defined owner. Only the owner can change ownership of a business view and specify which users and groups can read, change, delete and/or control a business view. Permissions are created and modified in the Business View Explorer.

Ownership

The creator of the business view is the default owner.

To change ownership, do the following:

1. Open the Business View Explorer by either clicking on its icon or by using the menu toolbar (*Tools/Business View Explorer*). See Figure above.
2. Click **Show Permissions**  icon.
3. Click on the business view you want to change.
4. Click the **Change** button next to *the Owner* field. The following screen is displayed.

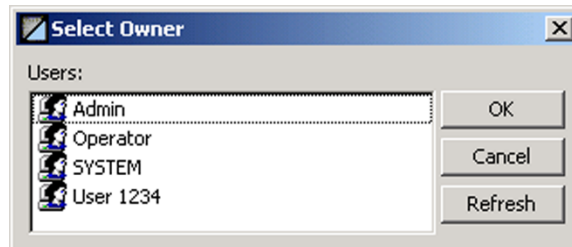


Figure 4-67. Selecting Ownership of Business View

5. Select an *Owner* from the *Users* column and click **OK**.

Add/Remove Users and Groups

To add a User and/or Group, do the following:

1. Click the **Add** button next to the Users and Groups field. The following screen is displayed.



Figure 4-68. Selecting Users or Groups Permission

2. Select user or group and click **OK**.

To remove a User and/or Group, highlight the user or group to be removed and click **Remove**.

Modify Access

To modify access by a user or group, select *Read*, *Change*, *Delete* and/or *Control*.

- *Read* – user may only read/view the business view.
- *Change* – user may change any of the attributes of the business view.
- *Delete* – user may delete the business view.

- *Control* – user may execute control actions on the business view such as start, stop, or disable.

4.8.3 Business View Sensors

Sensors are the building blocks of every business view. They are organized into a hierarchical structure and feed one another with status and real-time events.

Sensors are used to monitor one or more facts. Sensors have the ability to enumerate facts at run-time and determine which facts need to be evaluated. It works very much like a FOR or WHILE loop on a variable list of facts. (Example: A difficulty in monitoring peak response time in all servlets within IBM MQ is that the number of servlets could be large and may change at run-time.)

Sensors act like probes that evaluate facts or other sensors. Each sensor cycles through the following run-time stages:

- Receives inputs from facts or other child sensors
- Evaluates inputs by applying user-defined logical rules
- Assigns severities to (source) fact/sensor input based on user-defined Sensor parameters
- Triggers an alert or an action as defined by the user
- May log its information to a log file or relational database

4.8.3.1 Sensor Categorization

Sensors can be categorized as follows and are configured in *sensor properties* under the *General* tab:

- **Service category** – classification of the monitored service. (Refer to [Table 4-28](#).)
- **Service type** – name or identity of the monitored service.
- **Object type** – specific name of the resource.

4.8.3.2 Sensor Tables

The sensor database table layout below lists all the sensors along with the total number of allocated characters for each (if applicable).

Table 4-27. Sensor Database Layout	
Sensor	Total Characters/Value
ManagerName	varchar(128)
PolicyName	varchar(128)
SensorName	varchar(255)
Severity	varchar(32)
SensorValue	varchar(255)
NumericValue	float

Table 4-27. Sensor Database Layout	
Sensor	Total Characters/Value
Health	float
LogTime	timestamp
EventID	integer
ServerName	varchar(255)
OSName	varchar(128)
OSVersion	varchar(12)
OSArch	varchar(12)
OSUser	varchar(48)
APUser	varchar(48)
ServiceCategory	integer
ServiceType	varchar(48)
ObjectType	varchar(48)

The Service Category Table is defined below.

Table 4-28. Service Category	
Service Category Number	Service Category Name
0	Hardware
1	Network
2	Server
3	Operating System
4	Middleware
5	Database
6	Application Server
7	Web Server
8	Web Service
9	Client
10	Application
11	IT Service
12	Business Service
13	Transaction
14	Policy
15	Miscellaneous
16	Other

4.8.3.3 Sensor Scripts

Scripts are used in the Sensors to define specified user actions and schemes. The Sensor Wizard is used to graphically select and define rules. The wizard generates the appropriate script automatically, which could be edited manually in advanced mode. Using advanced mode is recommended only for users familiar with sensor scripting language.

4.8.3.4 Fact Volatility

Volatility refers to the standard deviation of the change in value over time of a certain variable. Standard deviation is a measure of the dispersion of a set of values. It is defined as the root-mean-square (RMS) deviation of the values from their mean, or as the square root of the variance. If many data points are close to the mean, the standard deviation is small; if many data points are far from the mean, the standard deviation is large. If all data values are equal, the standard deviation is zero.

Fact volatility can be measured based on the standard deviation of the % change of the numerical value of the fact. Three fact parameters are relevant here:

- History-%Mean – mean based on % change
- History-% Variance – variance based on % change
- History-% Deviation – deviation based on % change.

For these facts to be displayed, history must be turned on by specifying Fact History Size and Fact History Time from the properties screen of the selected sensor. (Refer to [Configuring Policies](#), Fact Options tab, for information on specifying these parameters.)

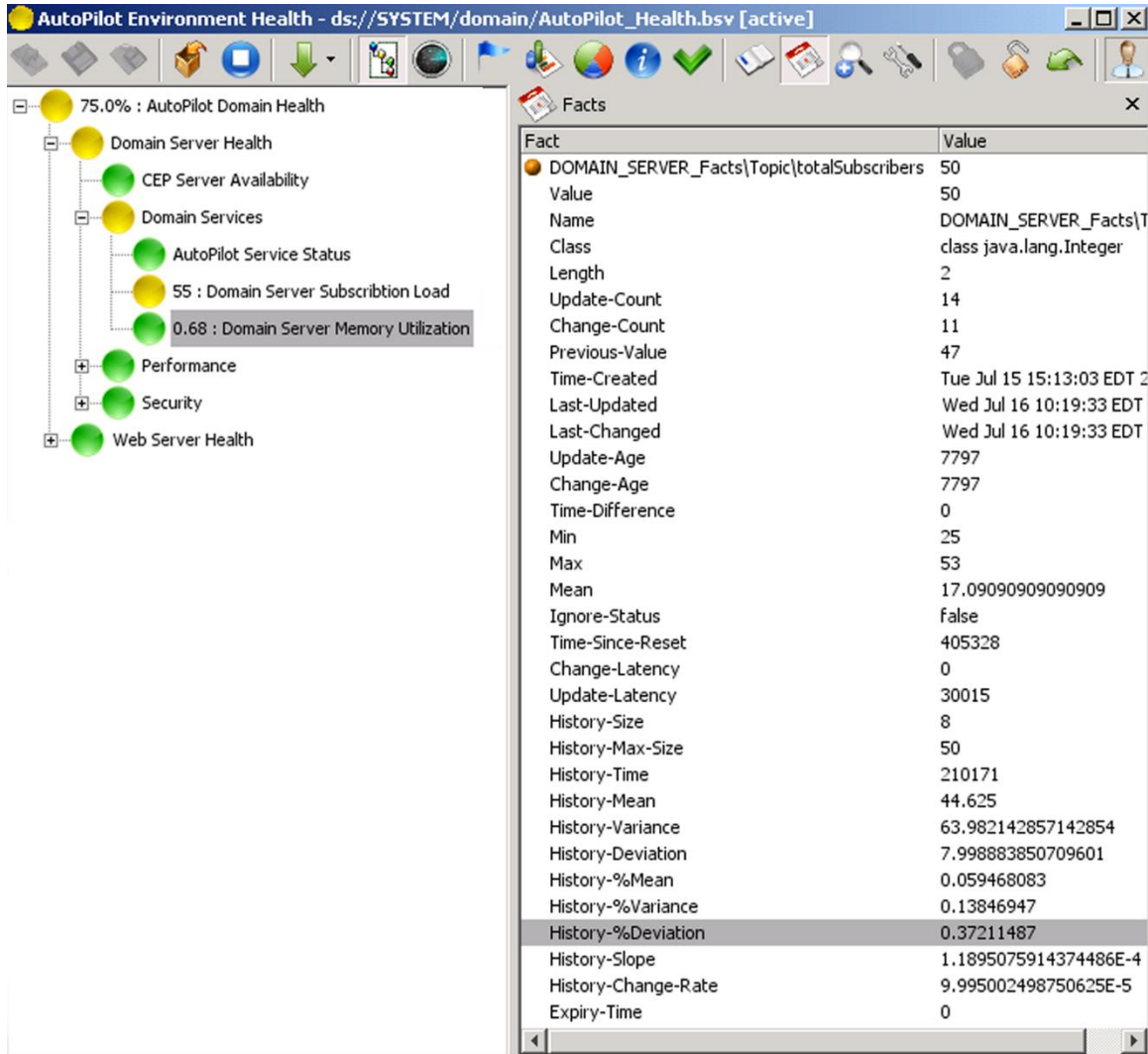


Figure 4-69. Fact Volatility

4.9 Alerts, Notifications and Rules

This section requires the use of *Business View Tool*. The tool can be accessed through the User Console.



TIP

When using Sensor Wizard, it is recommended that you make use of environment variables (specified in "Environment" options for each sensor). Environment variables can be used within the wizard in the form {var}. For example, instead of `DOMAIN_SERVER_Facts\Java\free_memory`, one can specify `{SERVER}_Facts\Java\free_memory`. This makes business views easier to manage and maintain and removes hard coded variables. This is especially useful when creating business views for deployment in multiple environments such as TEST, QA, and Production. Environment variables can be defined either within business views or `node.properties`. (Refer to [Server Runtime – PROPERTY FILES](#)).

4.9.1 Configuring Alerts

Alerts are normally set while you are building business views. The alert determines the severity level that triggers the alert and the method of notification. Alerts can be set while you are building business views or can be configured during a sensor update or change. The severities are defined by the sensor health they represent. When selecting alert severities, ensure that the health percentages and weight reflect your needs. Configure alerts as follows:

1. Open the effected business view.
2. Stop the business view if it is running by clicking on the **Stop** button.
3. Right-click the sensor to be changed. The sub-menu is displayed.



NOTE

If you want to make the same changes to two or more sensors, you can do so. Press **Shift** to select multiple sensors. Then make your changes on the appropriate **Properties** tab as described below.

4. Click **Properties**. The sensor properties for the selected sensor will be displayed on the right.

5. Click the **Alert** tab to access the alert properties.

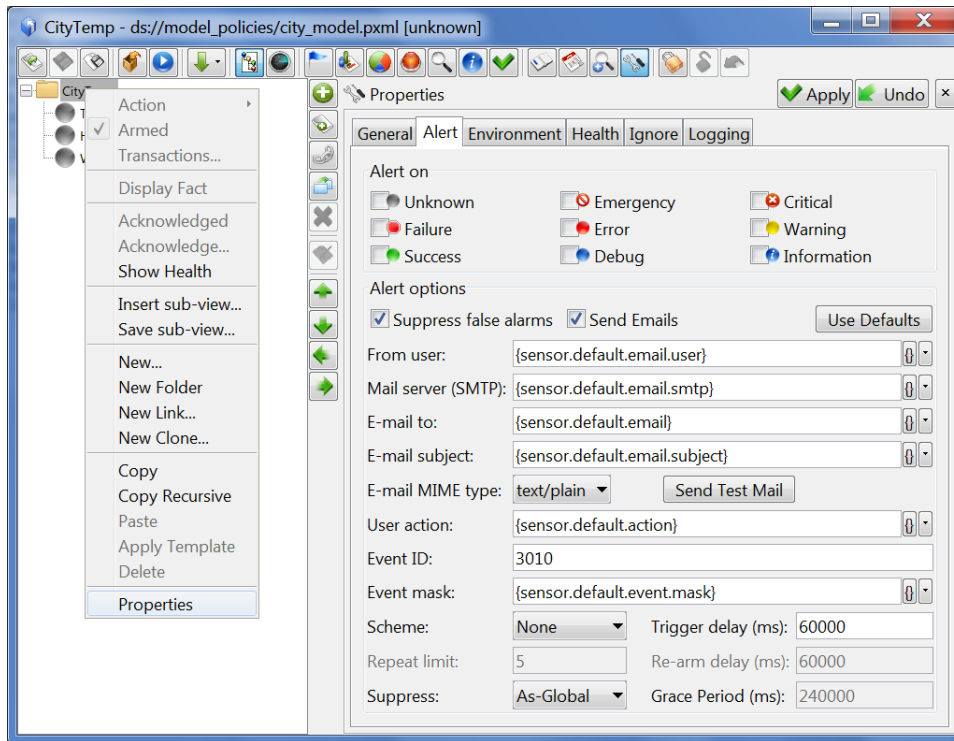


Figure 4-70. Changing Sensor Properties

6. Set the properties for your sensor alert using the options outlined in the table and figure below:

Table 4-29. Alert Properties

Parameter	Description
Severities: (Alert On)	Select the severity levels that will trigger notifications: <ul style="list-style-type: none"> Unknown: 0.0 (0%), default alert setting Warning: 0.75 (75%) Emergency: 0.1 (10%) Success: 1.0 (100%) Critical: 0.25 (25%) Debug: 1.0 (100%) Failure: 0.45 (45%) Information: 1.0 (100%) Error: 0.65 (65%)
Suppress false alarms	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> false alarm. Intelligent False Alarm suppression logic. Disables duplicate actions/notifications when sensors are acknowledged/armed.
Use Defaults	Click button to use default properties that were defined in domain.properties on each domain server installation. Click Yes in the confirmation box.
From User	Enter your mail server user id. Usually the same as your email address.

Table 4-29. Alert Properties

Parameter	Description
Mail Server (SMTP)	Enter the name of the local mail server (SMTP server).
E-mail to	Enter the email address of the party to be notified (e.g., %f0:user@meshIQ.com).
E-mail subject	Enter user-defined email subject line.
E-mail MIME type	Select text/plain or text/html email.
Send Test Mail	Click button to verify email reached the intended recipient.
User Action:	Use to specify an external command, shell script or executable: (e.g. <i>runaction.bat</i>). You may also pass business view context information that could be used within a script or action. Refer to Automated Actions for a complete list of supported action variables.
Event ID	User assigned event identification number. You can customize the number as needed. Use only numbers, no spaces (example: default = 3010. Custom = 4089). This ID is used to record events into the event log.
Event mask	Event format mask used to record events triggered by the sensor. Users may customize the event format. Refer to Automated Actions for a complete list of supported variables that can be used within the event mask.
Scheme	Open the Scheme menu and select the specific occurrence that will trigger notifications. None – Notification will not be generated. Once – Notification generated once per severity status change. All severities selected in Alert On section. Change – Notification generated with each change in status. Repeat – Sensor repeats actions/alerts until the sensor goes into a state which is not checked in the Alert On section. Sensor re-arms until the sensor goes back to an unchecked state. Notification frequency is based on Re-arm Delay intervals.
Trigger Delay	User-defined delay times expressed in milliseconds (1/1000th of a second). Default is 60000 or 60 seconds. The sensor will delay notification for the prescribed time.
Repeat Limit	Maximum action/notification repetitions.
Re-Arm Delay	User-defined re-arming delay of sensor notifications. The sensor will resample and resend alert notification continuously (per delay time) until the condition is corrected. Default is 60000 or 60 seconds.
Suppress	From the drop-down menu select one of the following: Off – all facts are considered regardless of how old they are As-Global – defined in global properties As-Parent – defined in parentage On-Change – considers change add as set in grace period On-Update – considers update age as set in grace period

Table 4-29. Alert Properties

Parameter	Description
Grace Period	Age of facts, in milliseconds, which will not be considered for evaluation.

7. Click **Apply** to complete sensor change.
8. Click **Start** to restart the business view.
9. When sensors are functional, alert options are in effect. Alerts/Email notifications will be transmitted as specified in your business view.

4.9.2 Defining Sensor Rules

A sensor is a run-time user-defined object that encapsulates data, rules, behavior, notifications, logging into a single component. Each sensor may evaluate one or more facts at once. All facts must be available (must have values) in order to complete rule evaluation. Rule evaluation halts when one or more facts are not bound (have no values). Facts without a value may occur in several situations: fact does not exist, or fact value is equal to null. Evaluation will resume on fact change or when fact value becomes available.

Sensors define the purpose and scope of one or more rules. Selected facts are the basis of what sensors monitor and under what conditions. Thresholds can be fixed values, other selected facts (variables) or environment variables defined in the business view, CEP server or Java environment.

Sensor rules are invoked on changes in the fact value and status. Each fact change is evaluated by the sensor rule and the result is mapped into any one of the supported severities such as SUCCESS, WARNING or CRITICAL. Automated actions, notification or logging can be triggered as a result of the rule evaluation.

Therefore, if the normal condition would indicate a status of "Success" the status would remain unchanged as long as the conditions for "Success" are maintained. If the condition changes to a condition outside of the "Success" parameters, the status is changed to reflect that condition. The change in status would change the existing severity status, thus triggering a predetermined action to attempt to correct or report the condition. When the condition returns to normal, the status is updated accordingly. If the "normal" condition or status remains unchanged then the sensor will report (typically) an "information" or "Success" status.

Example: if the purpose of the rule was to monitor the number of facts being monitored, then we would base the rule on the *totalFacts* fact. *totalFacts* monitors the current number of facts monitored by all services within the CEP server.

If the maximum number of facts that can be monitored is 500, you may want to be alerted if it reaches 400 or 80% capacity. By monitoring the current number of facts with a condition of "more than" and the variable condition set at "400" the status will change if the total

number of facts exceeds 400. The status will change to reflect the severity set in the rule. When the severity "Warning" is reached the rule triggers the action selected. In this sample, the action would be an audible beep. If the action you chose was to stop or update the fact, the associated fact in optional sensor/value would have been updated as specified.

Sensors have no predetermined limit to the number of rules that can be defined.



TIP

The "'" character is a reserved symbol and cannot be used as part of any name in meshIQ Platform Core Services.

4.9.2.1 Defining Sensor Logic

When a sensor is transitioning to a checked (alert is enabled) state, the following logic is performed:

- If a sensor's previous state has not completed its notification before a new sensor state is completed, the previous alarm is suppressed.
- If a sensor's previous state has not completed its notification before a new sensor state is completed, notification is posted on the new sensor *only* if the new state is more severe. The previous alarm will be suppressed.
- If the new state is less severe, notification is *not* posted and there is a false alarm condition. The previous alarm will be suppressed.

4.9.2.2 Creating Sensors with the Wizard (Non-Dynamic)

1. Open the business view to receive the sensor.

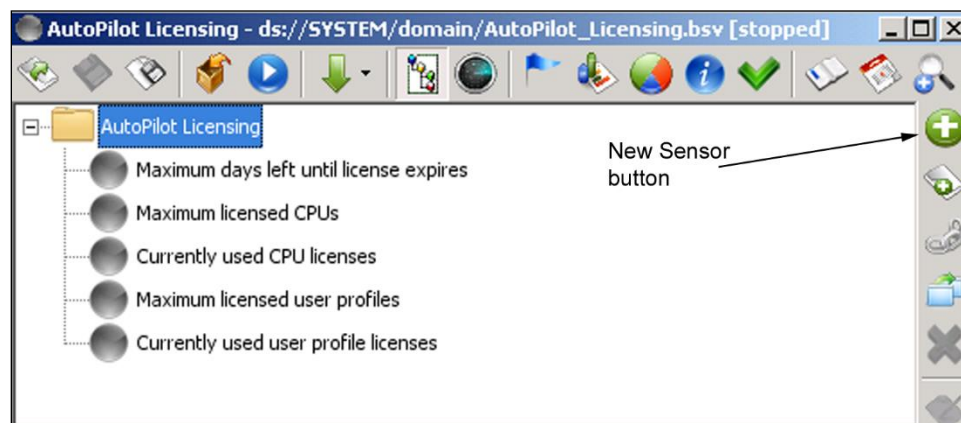


Figure 4-71. Add a New Sensor

2. Create a new sensor by clicking on the New Sensor button. The General Sensor Options screen is displayed.

Figure 4-72. Create a New Sensor - General Screen

3. Assign a logical sensor name in the *Name* field. The default is Untitled.
4. Identify the initial severity in the *Initial severity* field. The default is Unknown.
5. Add an initial value in the *Initial value* field. An initial sensor value is assigned when the sensor is initialized or when the fact is cleared. The initial value can be set for every sensor. Dynamic sensors inherit the initial value from the parent sensor. The default initial value is always set to null, unless modified by the user. Sensors will show the most recent value when the connection to the source or facts drops.
- 5a. Specify a numerical format, for example `###,###.##`.



The *Summarize* field only applies to dynamic sensors. Refer to Dynamic Sensors for an explanation of this field.

6. Add the specific name of the resource in the *Object type* field, for example: Queue Manager.
7. Add the name or identity of the monitored service in the *Service type* field, for example: IBM MQ.
8. Add the classification of the monitored service in the *Service category* field, for example: Server.
9. Provide a description of the sensor functionality and purpose.
If adding a hyperlink tag in the description field, do not use double quotes around the URL link.
For example, **use:** ` meshIQ`
Do not use: ` meshIQ`
10. Click **Next** to display the *Select Facts to Monitor* screen.

11. Select the desired fact; click **Include** to add the fact to the *Monitor these facts* list.
12. To specifically exclude a fact from monitoring, select the fact, and then click **Exclude**. The fact will be added to the *But not these facts* list.
13. If you want to convert hard coded names into names with environment variables, ensure that the *Auto parameterize* check box is selected and then click **Next**. The *Sensor Evaluation Rules* screen (Figure 4-74) will be displayed.

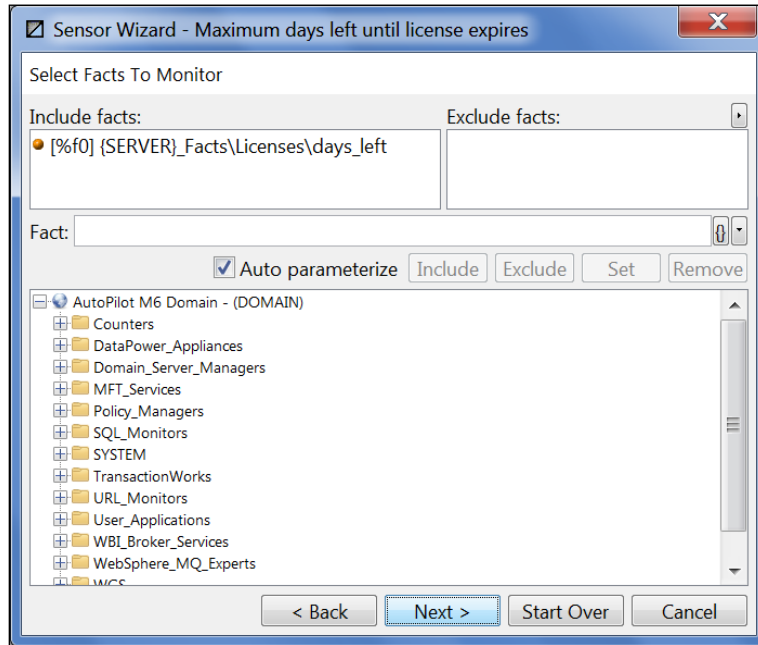


Figure 4-73. Selecting Facts to be Monitored


14. When you have added or excluded all required facts, click **Next** to proceed.


To improve sensor performance, the following type qualifiers can be added to values used within the sensors.

Table 4-30. Sensor Performance	
Type Qualifier	Description
\$int.	integer (example - \$int.100)
\$long.	long (example - \$long.100023)
\$float.	float
\$double.	double
\$bool.	Boolean
\$date.	date/time object
\$time.	date/time object
\$char.	string
\$systemtime.	system time in milliseconds

Supported Operators: The set of operators listed below can be used while defining the states of the sensor. These can be typed in or selected by clicking on the buttons shown below.

- + Adds the two operands on either side of the operator
- Subtracts the operand on the right from left
- * Generates a product of the two operands
- / Divides the operand on the left by the operand on the right.
- % Modulus operator - divides the operand on the left by the right and produces the remainder.


Click **Facts**  icon to insert the first fact in the list of those selected. Clicking on the newly inserted fact gives you the ability to select a different fact. Apart from the fact value you may also use other meta-data such as Last-Changed, Last-Updated, Length, Max, Mean, Min, etc. in your expression.

Click **Input**  icon to insert the numerical severity of the child sensor. (Useful only for static child sensors.)

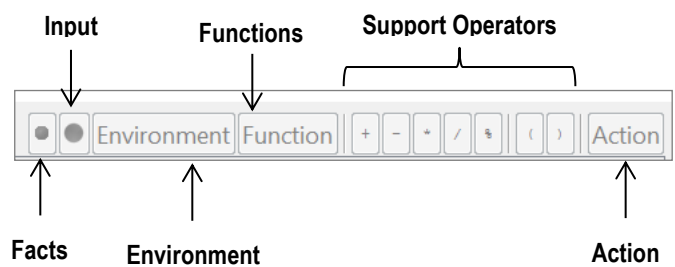
Click **Environment** to insert a reference to an environment variable 'user.name'. Clicking this variable makes a menu available for selecting other variables defined in the business view or globally in the different property files used by meshIQ Platform Core Services.

Click **Function** to insert a reference to a function variable 'abs()'. Clicking on this variable displays a menu for selecting other functions as defined in the business view or globally in the different property files used by meshIQ Platform Core Services.

Tab to the Action field and click **Action** button to insert a default action 'beep'. Clicking the newly inserted action allows you to select a fact and its first action as defined on the fact; you may now select different actions defined on the fact by clicking on the action displayed. You may also type in any action that you want executed if the sensor state gets set to the corresponding severity.



All the buttons shown below have been added to streamline the rule creation process. You may still copy from, paste to, or type directly into the field.



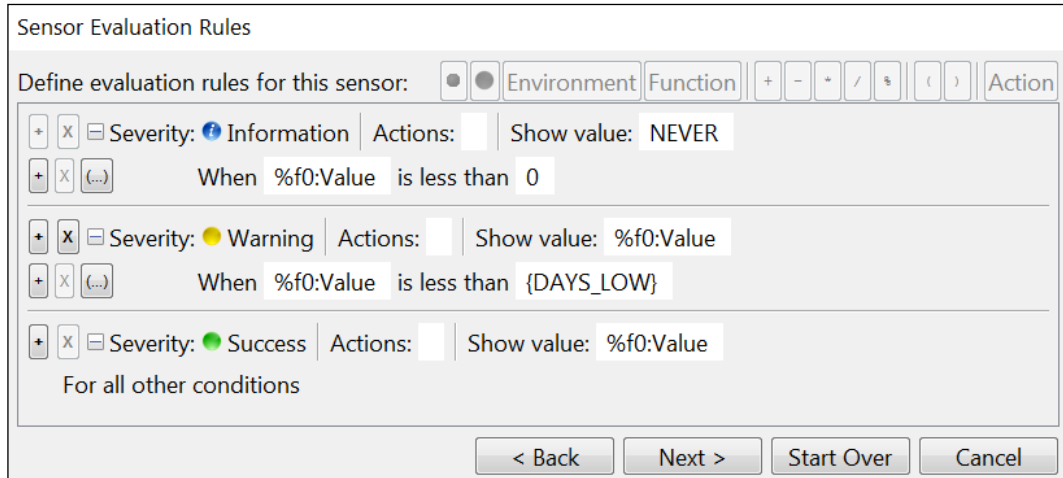





Figure 4-74. Sensor Rules

Rules

15. The  on adds an additional evaluation rule for determining sensor state and the  icon deletes the rule. The  icon allows the user to create nested “and/or” evaluation rules.

16. Select the fact to be evaluated from the facts included. All facts selected for use in this sensor will be listed in the menu. Facts are numbered (prefixed) as [%f0], [%f1] etc. Click the fact button on the upper-left side of the sensor screen to insert the first fact %f0. Metrics are derived from facts. There are two types of metrics: “out-of-the-box” and those based on history.

The screenshot shows the 'Sensor Wizard - Untitled 1' window. The 'Sensor Evaluation Rules' section contains two rules. The first rule is: 'When [%f0]:Value is greater than 0'. The second rule is: 'For all other conditions'. The 'Severities' list includes Unknown, Emergency, Critical, Failure, Error, Warning, Success, Debug, and Information. The 'Actions' list includes beep, halt, and [%f0] Workgroup_Policy_Manager\WMQ_Health_Policy\Detail\{server.grid.pri}. The 'Functions' list includes abs, CountAckedInputs, CountInputs, CountBelowSuccess, CountBelowWarning, CountUnknown, and CountEmergency. The 'Monitored Facts' list includes [%f0] Workgroup_Policy_Manager\WMQ_Health_Policy\Detail\{server.grid.pri}. The 'Condition' list includes is, is not, is greater than, is at least, is less than, is at most, matches, and does not match. The 'Context' list includes Value, Name, Location, Class, Length, Update-Count, Change-Count, Reset-Count, Anomaly-Count, and Anomaly-End-Count. Arrows point from the 'Severities' and 'Actions' lists to the 'Severity' and 'Actions' fields in the rules. An arrow points from the 'Monitored Facts' list to the '%f0:Value' field in the first rule. An arrow points from the 'Condition' list to the 'is greater than' field in the first rule. An arrow points from the 'Context' list to the 'abs()' field in the first rule.

Figure 4-75. Sensor Rule Inputs

17. Select the *conditional operator* to be used. The conditional operator is applied to the operands (value evaluated in the fields) on the left and right, and when the expression evaluates to true, the sensor is set to the specified severity.

- Use “greater than” or “less than” to determine if a fact value is within normal operational parameters.
- Use “matches” and “does not match” for comparison of string content.
 - a) Use %f0:Value matches AAA, to match the value if AAA is contained anywhere within.

- b) The placeholders of * and ? can be used where * matches any number of characters and ? matches any single character.

For example:

%f0:Value matches A*B?D matches anything that begins with A followed by any number of characters followed by B and then any character followed by D.

ABCD

ATTTTTTTTTTTBCD

Does not match is the opposite: it triggers if the object does not contain the matching text.

18. The value in the field selected may be evaluated against any value for the selected condition.
19. Select the severity level appropriate for the conditions being evaluated. Severity levels that reflect stoppages or failures can, and usually do, send notifications alerting the relevant personnel to the condition. In addition, the severity can trigger corrective actions to remedy the failure or diminished service.



NOTE

The default severity 'Unknown' is reserved and must be changed to an active severity level. If it remains unchanged you will not be able to proceed to the next screen.

20. Select an *action* that is appropriate for the severity reached. There can be multiple conditions and variables established for each fact monitored by using multiple rules, scripts, or user-defined commands. Hence there can be multiple severities and actions to support the needs of the sensor. Actions are triggered by the severity level reached. Some conditions will warrant no action, others will require multiple actions. The actions listed in the menu are numbered to reflect the fact number assigned when you include the fact(s) in the sensors. The predefined actions are repeated for each fact listed. The predefined actions listed are beep, halt and the list of facts included in the sensor.

You can specify an external command, shell script, or executable (example: *runaction.bat*). You may also pass business view context information that could be used within a script or action.

Users can manually enter actions by entering an assigned fact number (example: %f0.Start-Service) to initiate the specified action against the designated fact. See [Setting User Actions](#) for details on creating user actions and predefined actions.

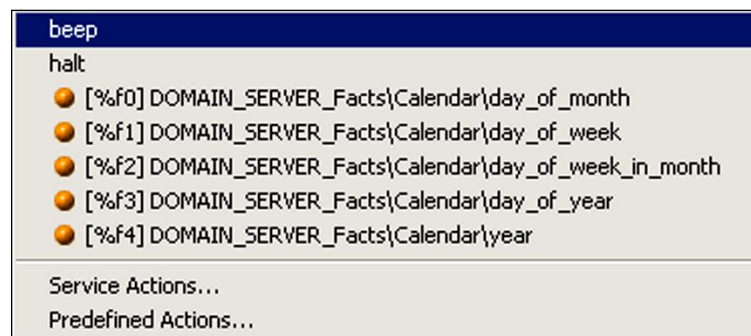


Figure 4-76. Actions

21. In addition, the *User Action* field on the **Define Alerts and Actions** tab can be used to specify an external command, shell script, or executable with environmental variables used as parameters of the command. (See Table 4-31, System Environment Variables). Click on the down arrow at the right end of the **User action** field to choose action choices as shown in the figure below.

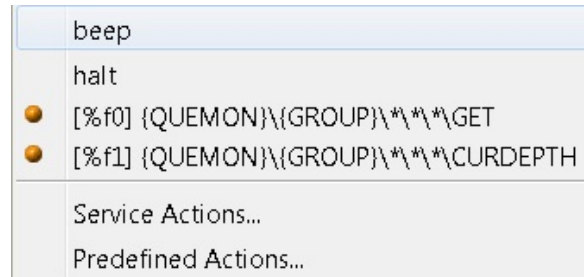


Figure 4-76-b. Sensor User Action Choices

They include:

- a simple beep
- halt
- a list of the sensor's facts: one more of the facts can be selected and inserted in the action as command parameters
- service actions: Built-in actions for each expert and policy manager, such as sending an SNMP V2 trap, see figure below.

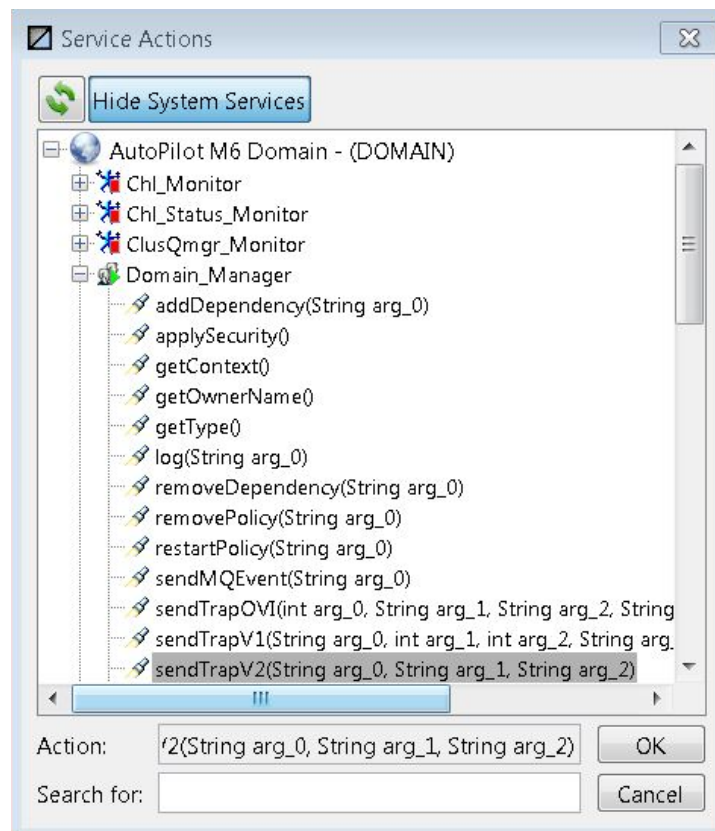


Figure 4-76-c. Service Actions

- pre-defined actions: Actions that users can define, such as a command to list the files in a directory (see the figure below).

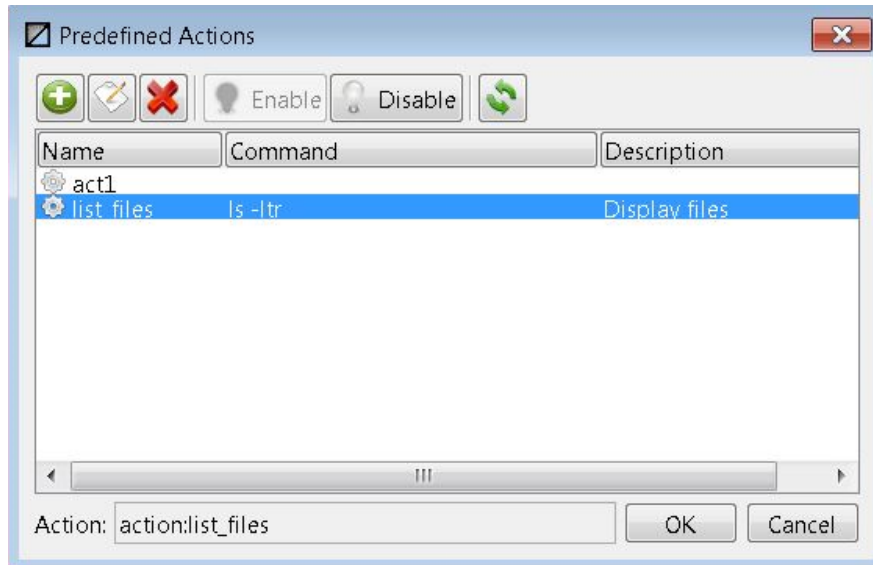


Figure 4-76-d. Predefined Actions

Example: `myscript.bat %sev% %desc% %facts% %event%`. A user action is usually taken in response to a problem or event.

If you want to receive an email notification of a specific event or condition, you can have the event, along with other status or properties of the sensor, included. Add action parameters to the *Action* to go with the script. To change the email behavior, add the following properties to the `[AUTOPILOT_HOME]/global.properties` file:

- `sensor.default.email.senderbehavior=group`
Can be set to one of the following:
 - **group:** All email recipients are shown (all email addresses in the “To” field of the email). For example, if the email is sent to three different email addresses, each recipient will see the other two recipients.
 - **individual:** Each recipient will only see themselves instead of the entire group.
- `sensor.default.email.smtp.connectiontimeout=15000`
This is the socket connection timeout value in milliseconds.
- `sensor.default.email.smtp.timeout=15000`
Socket read timeout value in milliseconds.

Table 4-31. System Environment Table Variables


Variable	Description
<code>%account%</code>	Name of current user context. Used within event mask and sensor action fields.
<code>%cause()</code>	Is an action that gets variable data from a child sensor that triggers the parent to go into alarm. Example: Root Bob Sally Joe

Table 4-31. System Environment Table Variables

Variable	Description
	<p>Chris</p> <p>If the parent, Bob, wants to generate the event, "My child has gone into alarm" but wants to include the child's name it could be done such as "My child %cause(%from%) has gone into alarm".</p> <p>Alternately, there is also parent() too, that is the children want to send the event but reference their parent. "I have gone into alarm, please notify %parent(%from%)"</p>
%date%	Date stamp of the event: Month, Date, Year
%desc%	Description of the firing sensor defined in the wizard or properties of the sensor.
%event%	Event message that qualifies the sensor. The message includes the same information that would be logged or viewable in an event viewer
%facts%	Facts and their values in the form: fact1=value1, fact2=value2,factN=value. The %facts% may need to be enclosed in "" quotes, since it may contain blanks or special characters.
%from%	Component that generated the event (sensor name in this case).
%f#(token)%	<p>Where %f# refers to the fact being monitored by a sensor and # is the fact index number, token is a zero-based token of the fact delimited by "\" delimiter.</p> <p>Example: Given %f0=Expert\TK1\TK2\TK3\TK3\Var</p> <p>Token1: TK1 as %f0(1)%, expert name: %f0(0)%, Token2: %f0(2)%</p> <p>Example: action %sevstr% token2="%f0(2)%"</p>
%health%	Sensor health index from 0.0 to 1.0
%id%	Message id of the sensor, as defined in the "Alert Id" field of the sensor "Alert" properties.
%ivalue%	The value of the child sensor that created the alert.
%parent%	Name of the immediate parent sensor.
%party%	List of email addresses parties as specified in <i>Email To</i> in the Alert Options.
%related%	Event message that triggered the sensor. May need to be enclosed in ""quotes, since it may contain blanks or special characters.
%root%	Name of the root sensor (topmost sensor) of the business view.
%sev%	Severity of sensor that fires the action; integer number from 0-8 representing sensor severities.
%sevstr%	Upper case string representation of %sev% integer code. Ex: WARNING, CRITICAL, SUCCESS.
%srvcaty%	Integer value of service category. Refer to Table 4-28 , Service Category.
%srvtype%	String value of service type.
%objtype%	String value of object type.
%time%	Time stamp of the event: HH:MM:SS AM/PM.

Table 4-31. System Environment Table Variables	
Variable	Description
%user%	Name of the user as specified in <i>From User</i> in the <i>Alert Options</i> .
%value%	Current sensor value as specified in the sensor wizard.
{env_var}	Where env_var is a java environment property. The complete list of properties varies from system to system.
%ovosev%	Maps Core Services severity to HPOVO severity.
%tecsev%	Maps Core Services severity to Tivoli TEC severity.

22. *Show Value* field defines the value for a sensor. This can be generated using the value of a fact directly or by evaluating an expression.



You may also use environment variables in the show value field.

23. Click **Next** and set properties for your *sensor alert* using options outlined in table and figure below:










Table 4-32. Alert Properties	
Parameter	Description
Severities:	Select the severity levels that will trigger notifications:  Unknown: 0.0 (0%), default alert setting  Emergency: 0.1 (10%)  Critical: 0.25 (25%)  Failure: 0.45 (45%)  Error: 0.65 (65%)  Warning: 0.75 (75%)  Success: 1.0 (100%)  Debug: 1.0 (100%)  Information: 1.0 (100 %)
Suppress false alarms	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> false alarm. Intelligent False Alarm suppression logic. Disables duplicate actions/notifications when sensors are acknowledged/armed.
Use Defaults button	Click Use Defaults to use the default properties that were defined in domain.properties on each domain server installation. Click Yes in the confirmation box.
From User	Enter your mail server user id. Usually is the same as your email address.
Mail Server (SMTP)	Enter the name of the local mail server (SMTP server).

Table 4-32. Alert Properties

Parameter	Description	
E-mail to	Enter the email address of the party to be notified (e.g., %f0:user@meshIQ.com).	
E-mail subject	Enter user-defined email subject line.	
E-mail MIME type	Select text/plain or text/html email	
Send Test Mail button	Click button to verify email reached the intended recipient.	
User Action:	Use to specify an external command, shell script or executable: (e.g., <i>runaction.bat</i>). You may also pass business view context information that could be used within a script or action. Refer to section 4.9.4, Automated Actions for a complete list of supported action variables.	
Event ID	User assigned event identification number. You can customize the number as needed, but only using numbers, no spaces (Example: default = 3010. Custom = 4089). This ID is used to record events into the event log.	
Event mask	Event format mask used to record events triggered by the sensor. Users may customize the event format. Refer to section 4.9.4, Automated Actions for a complete list of supported action variables that can be used within the event mask.	
Scheme	Open the Scheme menu and select the specific occurrence that will trigger notifications.	
	None: Notification will not be generated.	
	Once: Notification generated once per severity status change all severities selected in <i>Alert On</i> section.	
	Change: Notification generated with each change in status.	
Repeat Limit	Repeat: Sensor repeats actions/alerts until the sensor goes into a state, which is not checked in the "Alert On" section. Sensor re-arms until the sensor goes back to an unchecked state. Notification frequency is based on Re-arm Delay intervals.	
	Trigger Delay (ms)	User-defined delay times expressed in milliseconds (1/1000 th of a second). Default is 60000, or 60 seconds. The sensor will delay notification for the prescribed time.
	Repeat Limit	Maximum action/notification repetitions.
	Re-Arm Delay (ms)	User-defined re-arming delay of sensor notifications. The sensor will resample and resend alert notification continuously (per delay time) until condition is corrected. Default is 60000, or 60 seconds.

The screenshot shows a window titled "Sensor Wizard - AutoPilot Service Status" with a close button in the top right. The main area is titled "Define Alerts and Actions".

Alert on: A grid of checkboxes for alert levels:

- Unknown:
- Emergency:
- Critical:
- Failure:
- Error:
- Warning:
- Success:
- Debug:
- Information:

Alert options:

- Suppress false alarms
- Send Emails
- Use Defaults button

Fields:

- From user: {sensor.default.email.user}
- Mail server (SMTP): {sensor.default.email.smtp}
- E-mail to: {sensor.default.email}
- E-mail subject: {sensor.default.email.subject}
- E-mail MIME type: text/plain (dropdown), Send Test Mail button
- User action: {sensor.default.action}
- Event ID: 3010
- Event mask: {sensor.default.event.mask}
- Scheme: None (dropdown), Trigger delay (ms): 1000
- Repeat limit: 5, Re-arm delay (ms): 60000
- Suppress: As-Global (dropdown), Grace Period (ms): 240000

Buttons: < Back, Finish, Start Over, Cancel

Figure 4-77. Alert Properties

24. Click **Finish** to complete the sensor.


4.9.3 Dynamic Sensors

The dynamic sensor rules are defined and performed much like other sensors. The difference is those dynamic sensors are typically used to monitor the same fact used in multiple locations within the domain. For example, they can be used to monitor the status of multiple nodes in a meshIQ domain. To monitor the nodes' states you would have to verify that the fact reporting the state for all nodes was identical in name and function. By using a wildcard (* asterisk) in place of the node name in the address, you would include the state of every node in the domain. Once the sensor wizard recognizes the wildcard in an included or set fact, it identifies the sensor as dynamic.

For example: If the normal condition is all nodes are Active and a status of *Success* was defined for this state, then status would remain unchanged as long as all nodes were active. If one or more nodes stop, then the status would change to *Warning* or another condition defined in the rule.

The change in status would change the existing severity. The change could trigger a predetermined action to attempt to correct or report the change. When the condition returns to normal conditions the status is updated accordingly.

Notifications defined to alert you if a condition or node changes status would only alert you that there was a change, but not which node changed.



The forward slash “/” character is a reserved symbol and cannot be used as part of any name in meshIQ Platform Core Services.

Creating a Dynamic Sensor:

1. Open the business view to receive the dynamic sensor.

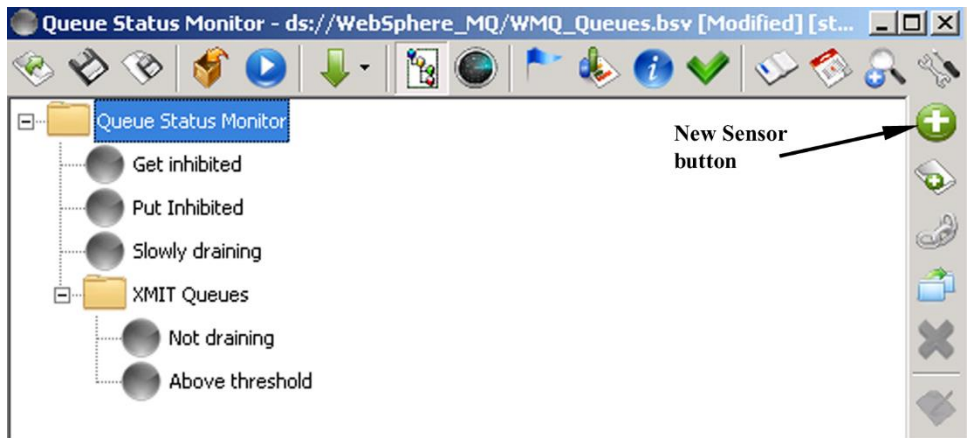


Figure 4-78. Adding New Dynamic Sensors

2. Create a new dynamic sensor by clicking on the **New Sensor** button. The *General Sensor Options* screen is displayed.

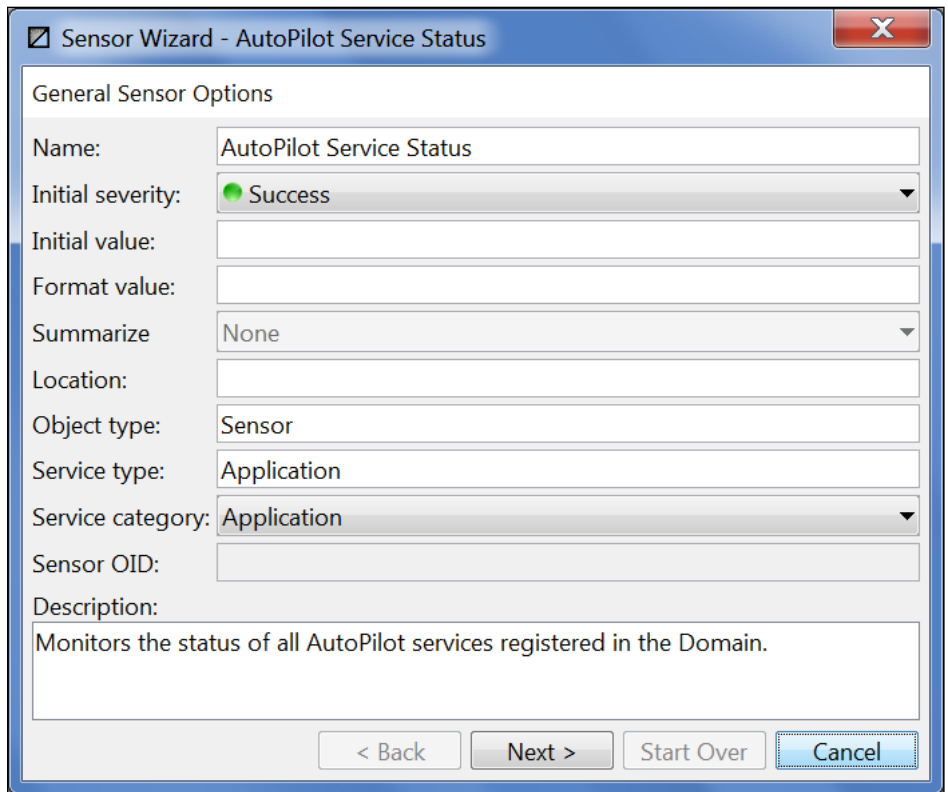


Figure 4-79. Sensor Name and Description

3. Assign a logical sensor name in the *Name* field. The default is Untitled.
4. Identify the initial severity in the *Initial severity* field. The default is Unknown.
5. Add an initial value in the *Initial value* field. An initial sensor value is assigned when the sensor is initialized or when the fact is cleared. The initial value can be set for every sensor. Dynamic sensors inherit the initial value from the parent sensor. The default initial value is always set to null, unless modified by the user. Sensors will show the most recent value when the connection to the source or facts drops.
6. The Summarize field is always disabled and will display the default value None. If a summarization method is displayed on the *Dynamic Sensors Options* screen (Step 18 below), this field will automatically change to display the summarization selected.
7. Add the specific name of the resource in the *Object type* field, for example: Queue Manager.
8. Add the name or identity of the monitored service in the *Service type* field, for example: IBM MQ.
9. Add the classification of the monitored service in the *Service category* field, for example: Server (Refer to [Table 4-28](#)).
10. Provide a description of the sensor functionality and purpose.
11. Click **Next** to display the *Select Facts to Monitor* screen.

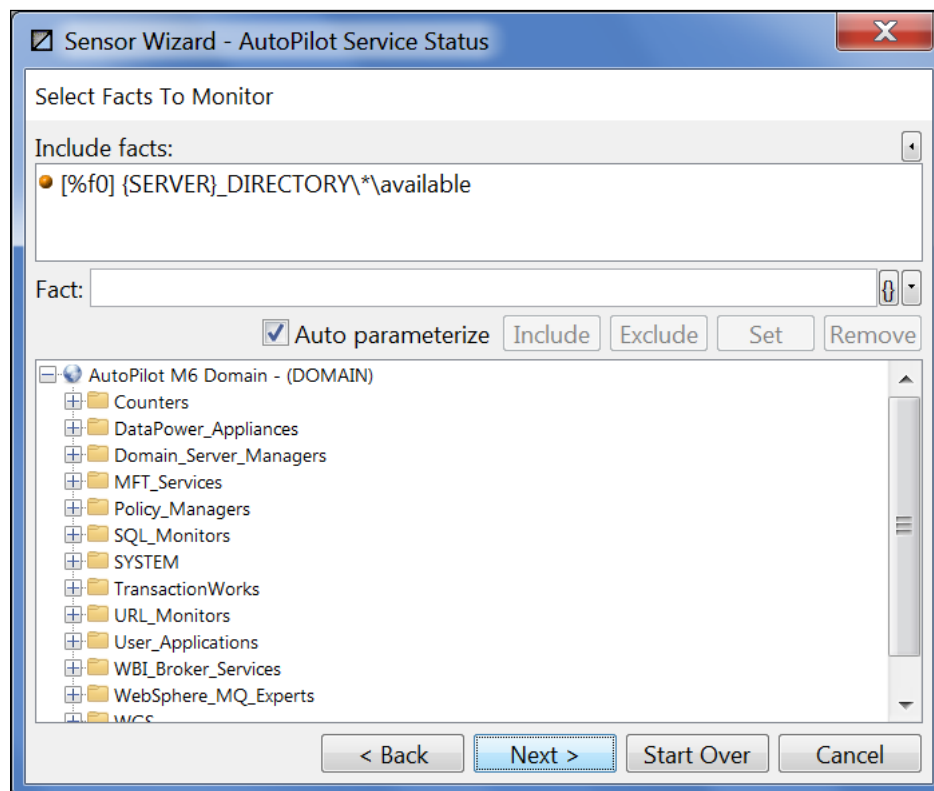


Figure 4-80. Sensor Wizard Fact Selection

12. Select the sensors that will be used to create the dynamic sensor.

13. Edit the sensor address by replacing the node name and any subsequent names (queue manager, queue, etc.) with a wild card (asterisk *). Using the wildcard will allow the sensor to be defined as a dynamic sensor and will monitor the state of all the nodes. Regexp can also be used (format %<regexp>).

Example: the sensors selected were

```
Que_Monitor\Workgroup\Node_Name]\Queue_Manager_Name]\ [Queue_Name]\
[fact].
```

The node name, queue manager name and queue name were each replaced with an * in order to monitor all nodes, queue managers and queues under Que_Monitor for current depth and usage. If the facts edited to include the wildcard are going to also be used for a child sensor, then you have to include these facts again to make them available.

Example for regexp:

```
OS_Monitor\%<A|B|C|>\value
```

Invalid regexp commands:

```
OS_Monitor\some text%<A|B|C|>\value
```

```
OS_Monitor\%<A|B|C|>some letters\value
```



The **Fact** field has search capability. Enter at least three characters and press **ctrl + space**. A drop-down list is displayed. Select a fact and click **Include** to include the selected fact.


NOTE

14. Click **Include** to include the selected fact(s).
15. If you want to convert hard coded names into names with environment variables, ensure that the *Auto parameterize* check box is selected. Click **Next** to display the *Create Dynamic Sensors* screen.


Supported Operators: The set of operators listed below can be used when you define the states of the sensor. These can be typed in or selected by clicking on the buttons shown below.

Supported Operators

- + Adds the two operands on either side of the operator
- Subtracts the operand on the right from left
- * Generates a product of the two operands
- / Divides the operand on the left by the operand on the right.
- % Modulus operator divides the operand on the left by the right and produces the remainder.

Click the **Facts**  icon to insert the first fact in the list of those selected. Clicking on the newly inserted fact gives you the ability to select a different fact. Apart from the fact value

you may also use other metadata such as Last-Changed, Last-Updated, Length, Max, Mean, Min, etc. in your expression.


Click **Input**  icon to insert the numerical severity of the child sensor. (Useful only for static child sensors.)

Click **Environment** to insert a reference to an environment variable 'user.name'. Clicking this variable makes a menu available for selecting other variables defined in the business view or globally in the different property files used by meshIQ Platform Core Services.

Click **Function** to insert a reference to a function variable 'abs()'. Clicking on this variable displays a menu for selecting other functions as defined in the business view or globally in the different property files used by meshIQ Platform Core Services.

Tab to the *Action* field and click the **Action** button to insert a default action 'beep'. Clicking on the newly inserted action allows you to select a fact and its first action as defined on the fact. You may now select different actions defined on the fact by clicking on the action displayed. You may also type in any action that you want executed if the sensor state gets set to the corresponding severity.

Select **Dynamic folders** to dynamically create folders in business views; for example, a folder named after a Queue Manager and containing alerts for all the channels belonging to the Queue Manager. This option is only for dynamic sensors.

 All the buttons shown below have been added to streamline the rule creation process. You may still copy from, paste to, or type directly into the field.

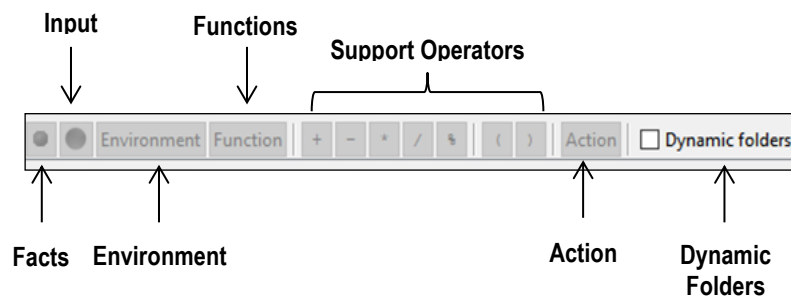



Figure 4-81. Supported Operators

 The Action support operator is not active for this screen. Actions are only applicable to the child sensors when using dynamic sensors.

16. Fact\context\conditions\variables:

- a. When: Select the dynamic fact you modified on the previous screen. The facts are numbered in the order they are included (example: [%f6]) .
- b. The context should reflect the nature of the fact and the intent of the sensor. *Value* is the default. In this sensor the default is used.
- c. Select the condition option to be used. The conditions give you a variable on which to base the sensor conditions. In this sample dynamic sensor, we used *is not*.
- d. Enter the user-defined value: fact/status/value, to finish defining the conditions. This sensor is monitoring the status (*Active* or *Stopped*, etc.). We want the sensor to report when any of the services are not active. You can use the operator to customize the variable to suit specific needs.

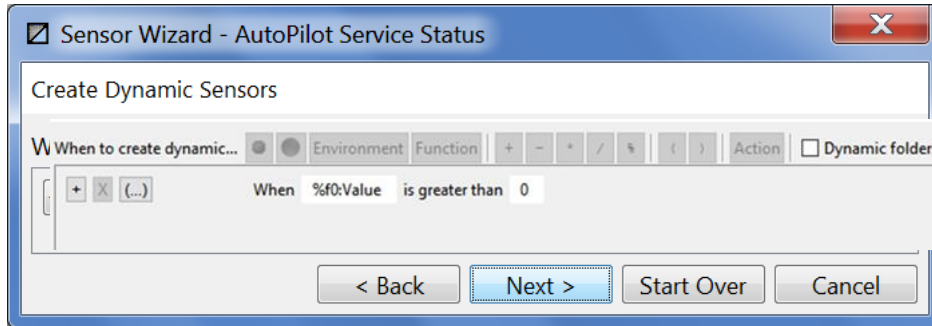


Figure 4-82. Creating Dynamic Sensors

- 17. Click **Next** to proceed to dynamic sensor options.
- 18. Define the sensor name and type as defined in the table below.

Table 4-33. Dynamic Sensor Properties		
Category	Property	Description
Dynamic sensor name	Use fact name	Uses the whole fact name as the Sensor name.
	Use part of fact name	Uses the first or user-defined segment of the fact name as the sensor name. Use directional arrows < > to move name to the desired segment.
	User-defined name	User-defined logical sensor name. Facts name can be tokenized using %#, where # is a zero-based token number. Example: Service %1 (recommended) See example and figure. Environment variables and system variables such as %sevstr% and %date% can be included as part of the user-defined name.
Dynamic sensor options	Summarize	Select how to summarize child sensors by selecting one of the following: Count; Count Acks; Max Value; Min Value; Average; Sum or Deviation. Only available at the dynamic sensor level. Default value is None.

Table 4-33. Dynamic Sensor Properties		
Category	Property	Description
	Expire dynamic sensor	If checked, enables expiration of the sensor as indicated by user in the seconds field. The sensor will disappear at a user-defined time limit, after reaching 100% health. The default is after 20 seconds, but it can be set to any duration needed. This is useful when corrected conditions need to be removed from the business view (recommended).

The dynamic sensor creates child sensors that monitor the facts at all the locations assumed in the wildcard. By maintaining them as a permanent sensor, they will log to the file or database as specified in the logging properties.

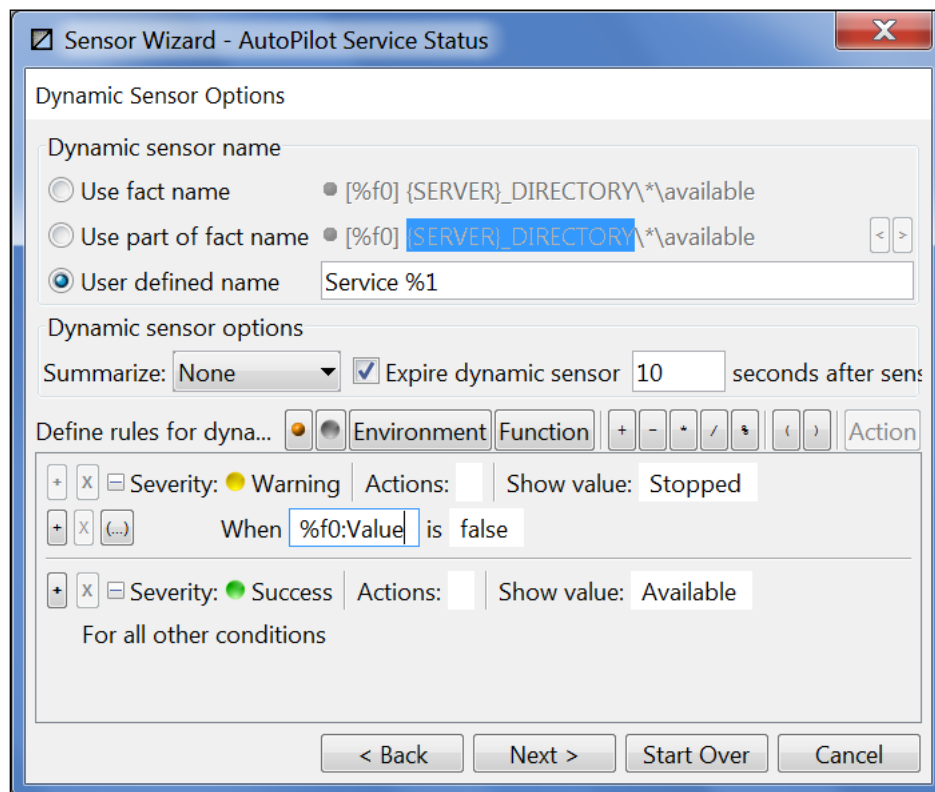


Figure 4-83. Dynamic Sensor Options

Dynamic Sensor Conditions: Define the sensor rules

19. The icon adds an additional evaluation rule for determining sensor state and the icon deletes the rule. The icon allows the user to create nested “and/or” evaluation rules.
20. Select the fact to be evaluated from the facts included. All facts selected for use in this sensor will be listed in the menu. Facts are numbered (prefixed) as [%f0], [%f1] etc. Click

the fact button on the upper-left side of the sensor screen to insert the first fact %f0. Metrics are derived from facts. There are two types of metrics: "out-of-the-box" and those based on history.

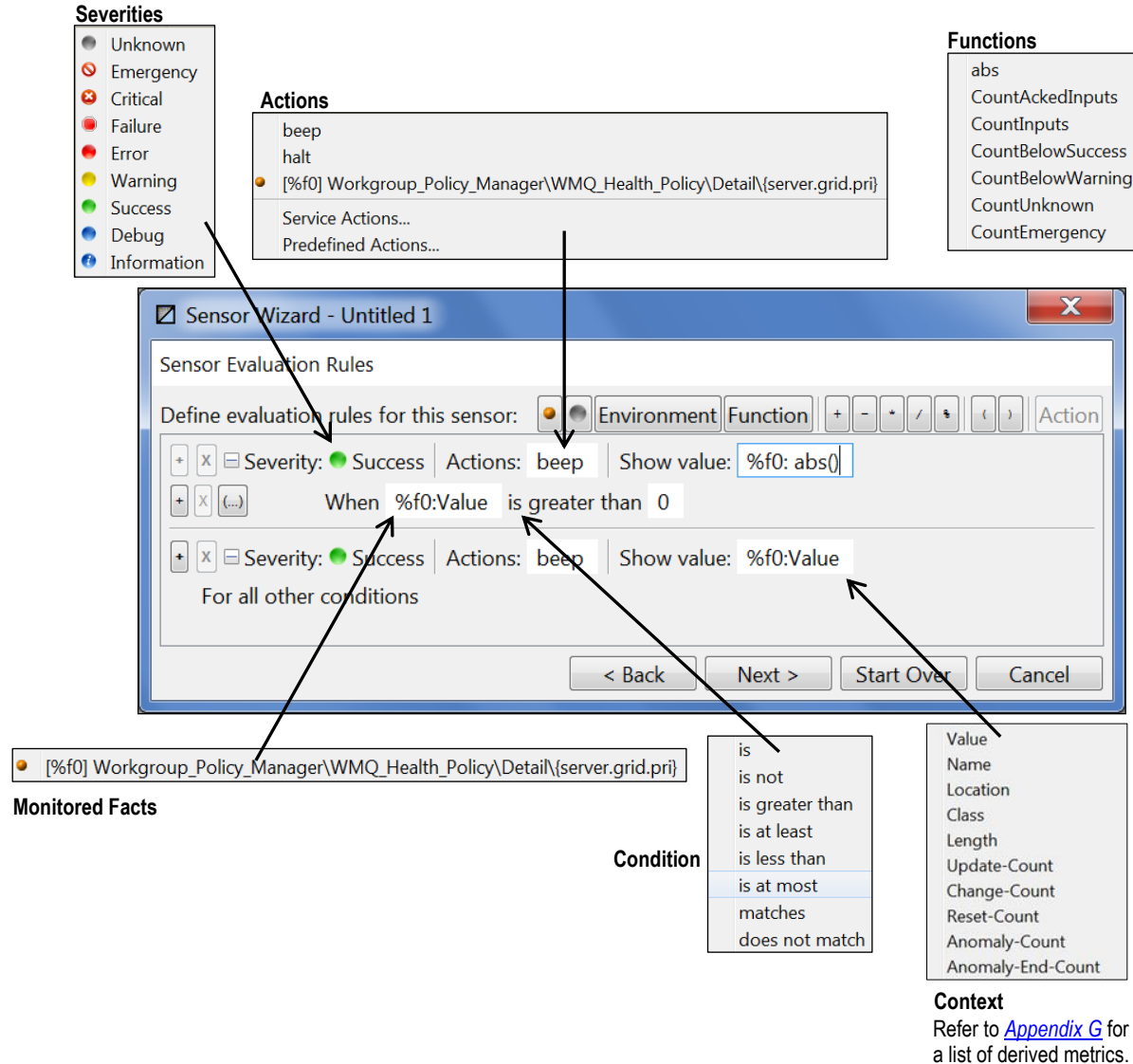


Figure 4-84. Sensor Rule Inputs

21. Select the *conditional operator* to be used. The conditional operator is applied to the operands (value evaluated in the fields) on the left and right, when the expression evaluates to true, the sensor is set to the specified severity. For example: using "greater than" or "less than" to determine if a fact value is within normal operational parameters. The value in the field selected may be evaluated against any value for the selected condition.
22. Select the severity level appropriate for the conditions being evaluated. Severity levels that reflect stoppages or failures can, and usually do, send notifications alerting the

relevant personnel to the condition. In addition, the severity can trigger corrective actions to remedy the failure or diminished service.



The default severity 'Unknown' is reserved and must be changed to an active severity level. If it remains unchanged you will not be able to proceed to the next screen.

23. Select an *action* that is appropriate for the severity reached. There can be multiple conditions and variables established for each fact monitored by using multiple rules or using scripts, user define commands. Hence there can be multiple severities and actions to support the needs of the sensor. Actions are triggered by the severity level reached. Some conditions will warrant no action, others will require multiple actions. The actions listed in the menu are numbered to reflect the fact number assigned when you include the fact(s) in the sensors. The pre-defined actions are repeated for each fact listed. The pre-defined actions listed are beep, halt and the list of facts included in the sensor, service actions and predefined actions. See similar section [Creating Sensors with the Wizard \(Non-Dynamic\)](#), step 21, for definitions and figures. You can specify an external command, shell script or executable (e.g. *runaction.bat*). You may also pass business view context information that could be used within a script or action.

Optionally, users can manually enter actions by entering the assigned fact number to initiate the specified action against the designated fact.

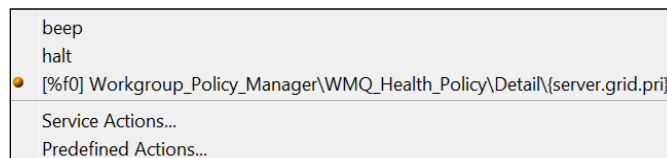


Figure 4-85. Actions

In addition, the *User action* field on the **Define Alerts and Actions** tab (see figure below) can be used to specify an external command, shell script or executable (for example: *myscript.bat %sev% %desc% %facts% %event% etc.*) (see table 4-31 System Environment Variables). The *User action* is usually taken in response to a problem or event. If you want to receive an email notifying you of a specific event or condition, you can have the event, along with other status or properties of the sensor included by adding action parameters to the *Action* to go with the script.

See [Setting User Actions](#) for details on creating user actions and predefined actions.

The screenshot shows a window titled "Sensor Wizard - Service Restart and Recovery" with a close button in the top right. The main heading is "Define Alerts and Actions".

Alert on: A grid of radio buttons for alert levels: Unknown, Failure, Success, Emergency, Error, Debug, Critical, Warning, and Information. The "Warning" option is selected.

Alert options: Includes checkboxes for "Suppress false alarms" and "Send Emails", and a "Use Defaults" button.

Fields: From user, Mail server (SMTP), E-mail to, E-mail subject, E-mail MIME type (set to text/plain), User action (method:{server.container.name}_SYSTEM.setMonitorState(%f0(2)%,true), Event ID (3010), Event mask, Scheme (None), Trigger delay (ms) (5000), Repeat limit (5), Re-arm delay (ms) (5000), Suppress (On-Change), and Grace Period (ms) (240000).

Buttons: < Back, Finish, Start Over, Cancel.

Figure 4-85-b. Policy CEP Server Health Sensor Wizard – Define Alerts and Actions


Table 4-34. System Environment Variables

Variable	Description
%account%	Name of current user context. Used within event mask and sensor action fields.
%cause()	<p>Is an action that gets variable data from a child sensor that triggered the parent to go into alarm. Example:</p> <pre>Root Bob Sally Joe Chris</pre> <p>If the parent, Bob, wants to generate the event, "My child has gone into alarm" but wants to include the child's name it could be done such as "My child %cause(%from%) has gone into alarm".</p> <p>Alternately, there is also parent() too, that is the children want to send the event but reference their parent. "I have gone into alarm, please notify %parent(%from%)"</p>
%date%	Date stamp of the event: Month, Day, Year
%desc%	Description of the firing sensor defined in the wizard or properties of the sensor.
%event%	Event message that qualifies the sensor. The message includes the same information that would be logged or viewable in an event viewer.
%facts%	Facts and their values in the form: fact1=value1,fact2=value2,factN=value. The %facts% may need to be enclosed in "" quotes, since it may contain blanks or special characters.
%from%	Component that generated the event (sensor name in this case).
%f#(token)%	<p>Where %f# refers to the fact being monitored by a sensor and # is the fact index number, token is a zero-based token of the fact delimited by "\" delimiter.</p> <p>Example: Given %f0=Expert\TK1\TK2\TK3\TK3\Var Token1: TK1 as %f0(1)%, expert name: %f0(0)%, Token2: %f0(2)% Example: action %sevstr% token2="%f0(2)%"</p>
%health%	Sensor health index from 0.0 to 1.0
%id%	Message id of the sensor, as defined in the <i>Alert Id</i> field of the sensor <i>Alert</i> properties.
%parent%	Name of the immediate parent sensor.
%party%	List of email addresses parties as specified in the <i>Email To</i> field in the <i>Alert</i> properties.
%related%	Event message that triggered the sensor. May need to be enclosed in "" quotes, since it may contain blanks or special characters.
%root%	Name of the root sensor (topmost sensor) of the business view.

Table 4-34. System Environment Variables

Variable	Description
%sev%	Severity of the sensor that fires the action; integer number from 0-8 representing sensor severities.
%sevstr%	Upper case string representation of the %sev% integer code. Example: WARNING, CRITICAL, SUCCESS.
%srvcaty%	Integer value of service category. Refer to Table 4-28 , Service Category.
%srvtype%	String value of service type.
%objtype%	String value of object type.
%time%	Time stamp of the event: HH:MM:SS AM/PM
%user%	Name of the user as specified in the <i>From User</i> field of the sensor <i>Alert</i> properties.
%value%	Current sensor value as specified in the sensor wizard.
{env_var}	Where env_var is a java environment property. The complete list of properties varies from system to system.
%ovosev%	Maps Core Services severity to HPOVO severity.
%tecsev%	Maps Core Services severity to Tivoli TEC severity.

24. *Show value* field defines the value for a sensor. This can be generated using the value of a fact directly or by evaluating an expression.



You may also use environment variables in the show value field.

25. Set the properties for your *sensor alert* using the options outlined in the table and figure below:

Table 4-35. Alert Properties

Parameter	Description
Severities:	Select the severity levels that will trigger notifications: <ul style="list-style-type: none"> <input type="checkbox"/> Unknown: 0.0 (0%), default alert setting <input checked="" type="checkbox"/> Emergency: 0.1 (10%) <input checked="" type="checkbox"/> Critical: 0.25 (25%) <input checked="" type="checkbox"/> Failure: 0.45 (45%) <input checked="" type="checkbox"/> Error: 0.65 (65%) <input checked="" type="checkbox"/> Warning: 0.75 (75%) <input checked="" type="checkbox"/> Success: 1.0 (100%) <input checked="" type="checkbox"/> Debug: 1.0 (100%) <input checked="" type="checkbox"/> Information: 1.0 (100 %)

Table 4-35. Alert Properties

Parameter	Description
Suppress false alarms	Enable/Disable <input checked="" type="checkbox"/> / <input type="checkbox"/> false alarm. Intelligent False Alarm suppression logic. Disables duplicate actions/notifications when sensors are acknowledged/armed.
Use Defaults	Click Use Defaults to use the default properties that were defined in domain.properties on each domain server installation. Click Yes in confirmation box.
From User	Enter your mail server user id. Usually is the same as your email address.
Mail Server (SMTP)	Enter the name of the local mail server (SMTP server).
E-mail to	Enter the email address of the party to be notified (e.g., %f0:user@meshIQ.com)
E-mail subject	Enter user-defined email subject line.
E-mail MIME type	Select text/plain or text/html email.
Send Test Mail	Click button to verify email reached the intended recipient.
User Action:	Use to specify an external command, shell script or executable: (e.g., <i>runaction.bat</i>). You may also pass business view context information that could be used within a script or action. Refer to section 4.9.4, Automated Actions for a complete list of supported action variables.
Event ID	User assigned event identification number. You can customize the number as needed, but only using numbers, no spaces (example: default = 3010. Custom = 4089). This ID is used to record events into the event log.
Event mask	Event format mask used to record events triggered by the sensor. Users may customize the event format. Refer to section 4.9.4, Automated Actions for a complete list of supported action variables that can be used within the event mask.
Scheme	Open the Scheme menu and select the specific occurrence that will trigger notifications.
	None: Notification will not be generated.
	Once: Notification generated once per severity status change all severities selected in Alert On section.
	Change: Notification generated with each change in status.
Trigger Delay	Repeat: Sensor repeats actions/alerts until the sensor goes into a state which is not checked in the Alert On section. Sensor re-arms until the sensor goes back to an unchecked state. Notification frequency is based on Re-arm Delay intervals.
	User-defined delay times expressed in milliseconds (1/1000th of a second). Default is 60000, or 60 seconds. The sensor will delay notification for the prescribed time.
Repeat Limit	Maximum action/notification repetitions.

Table 4-35. Alert Properties	
Parameter	Description
Re-Arm Delay	User-defined re-arming delay of sensor notifications. The sensor will resample and resend alert notification continuously (per delay time) until condition is corrected. Default is 60000, or 60 seconds.

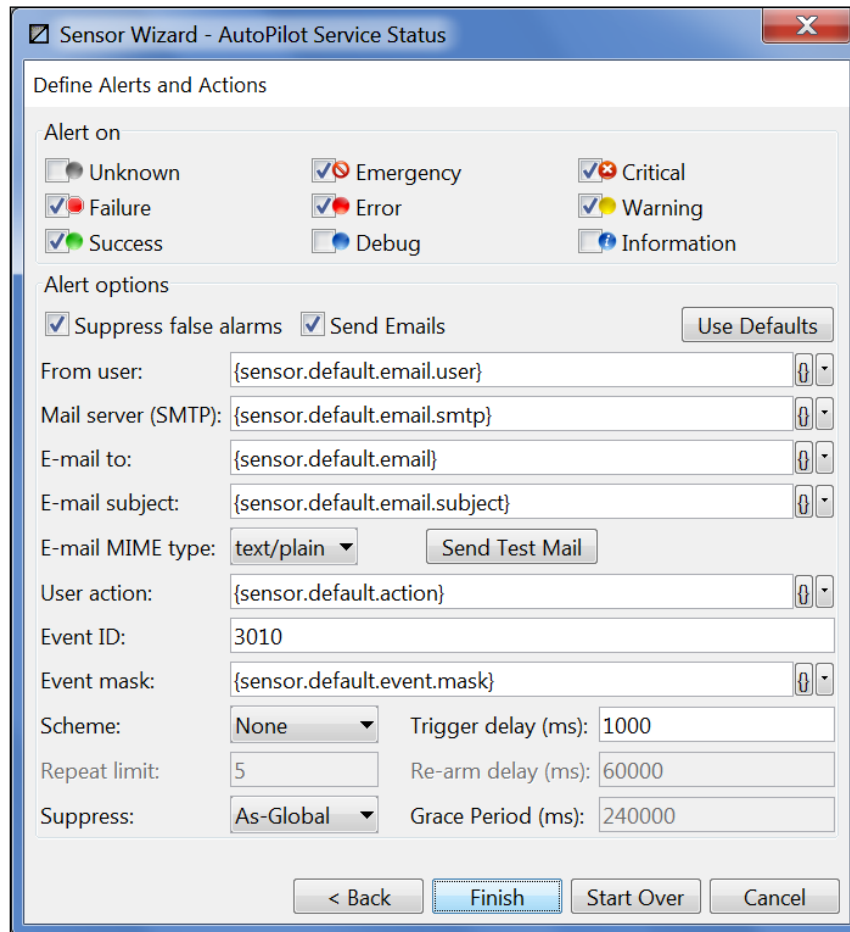


Figure 4-86. Alert Properties

26. Click **Finish** to complete the sensor.

4.9.4 Automated Actions

The sensors in business views monitor one or more facts; based on the rules applied and conditions the sensor will react in a manner you prescribed when you defined the sensor.

By using user actions to invoke automated responses, you can overcome system conditions and failures that would interfere with your operations or process.

Sensors evaluate inputs by applying user-defined (logical) rules. It assigns Severities to (source) Fact/Sensor inputs based on user-defined sensor parameters.

A sensor evaluates the rules and assigns severity when the values of the monitored facts change. It will initiate a number of user-defined actions as well. It may trigger Alerts

(example: beeps, email, pager messaging etc.). It can also start automated user scripts within the sensor (example: start procedures or redirecting data to a new collection point).



The Scheme setting controls both "User action" and email notifications.

Sensor *User Action* fields can be used to specify an external command, shell script or executable (example: `myscript.bat %sev% %desc% %facts% %event%`). The User action is usually taken in response to a problem or event. If you want to receive an email notifying you of a specific event or condition, you can have the event, along with other status or properties of the sensor included by adding action parameters to the User Action to go with the script.

Table 4-36. User Action Parameters

Variable	Description
%account%	Name of current user context. Used within event mask and sensor action fields.
%date%	Date stamp of the event: Month, Day, Year
%desc%	Description of the firing sensor defined in the wizard or properties of the sensor.
%event%	Event message that qualifies the sensor. The message includes the same information that would be logged or viewable in an event viewer.
%facts%	Facts and their values in the form: <code>fact1=value1, fact2=value2, factN=value</code> . The %facts% may need to be enclosed in "" quotes, since it may contain blanks or special characters.
%from%	Component that generated the event (sensor name in this case).
%f#(token)%	Where %f# refers to the fact being monitored by a sensor and # is the fact index number, token is a zero-based token of the fact delimited by "\" delimiter. Example: Given %f0=Expert\TK1\TK2\TK3\TK3\Var Token1: TK1 as %f0(1)%, expert name: %f0(0)%, Token2: %f0(2)% Example: action %sevstr% token2="%f0(2)%"
%health%	Sensor health index from 0.0 to 1.0
%id%	Message id of the sensor, as defined in the <i>Alert ID</i> field of the sensor <i>Alert</i> properties.
%parent%	Name of the immediate parent sensor.
%party%	List of email addresses parties as specified in <i>Email To</i> in the <i>Alert</i> Options.
%related%	Event message that triggered the sensor. May need to be enclosed in "" quotes, since it may contain blanks or special characters.
%root%	Name of the root sensor (topmost sensor) of the business view.

Table 4-36. User Action Parameters	
Variable	Description
%sev%	Severity of the sensor that fires the action; integer number from 0-8 representing sensor severity.
%sevstr%	Upper case string representation of the %sev% integer code. Example: WARNING, CRITICAL, SUCCESS.
%srvcaty%	Integer value of service category. Refer to Table 4-28 , Service Category.
%srvtype%	String value of service type
%objtype%	String value of object type
%time%	Time stamp of the event: HH:MM:SS AM/PM
%user%	Name of the user as specified in <i>From User</i> in the <i>Alert Options</i>
%value%	Current sensor value as specified in the sensor wizard.
{env_var}	Where env_var is a java environment property. The complete list of properties varies from system to system.
%ovosev%	Maps Core Services severity to HPOVO severity.
%tecsev%	Maps Core Services severity to Tivoli TEC severity.

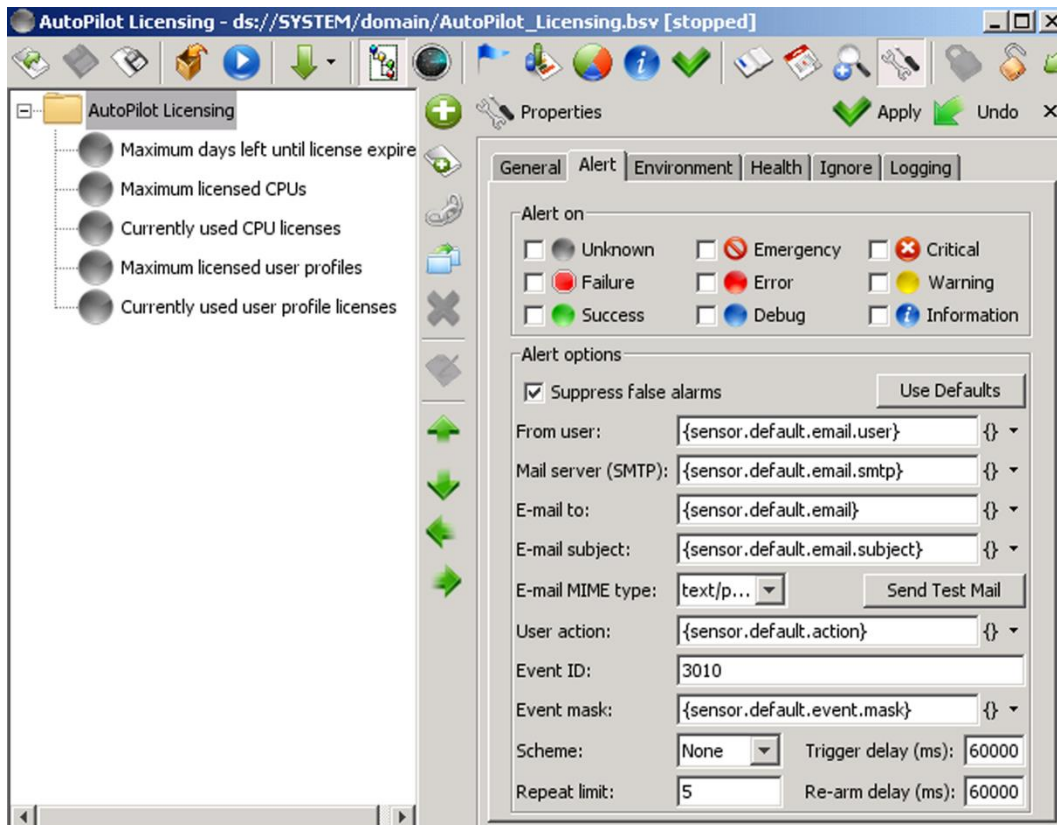



Figure 4-87. Sensor Alert Options

4.9.5 Managing Sensors

The user can add, view, change, copy, and debug sensors by using the sensor options menu.


NOTE
The availability of some options depends on the status of the policy.

Right-click a selected sensor to display the sensor options menu.

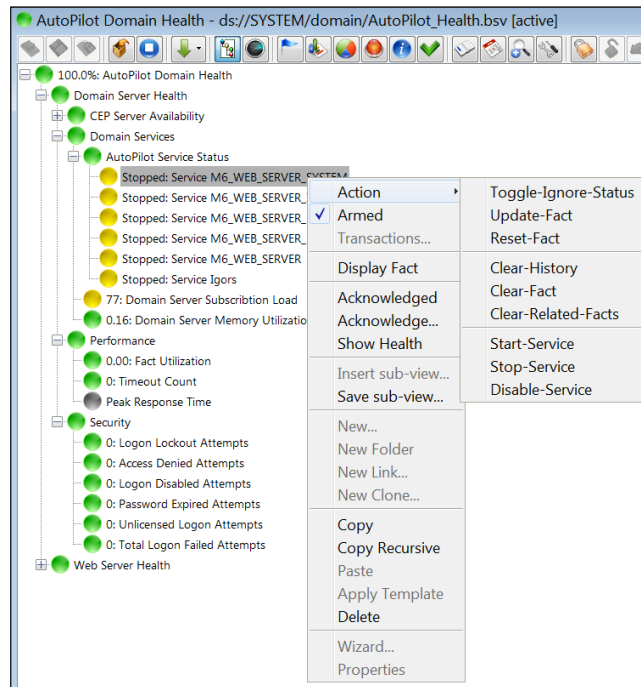


Figure 4-88. Sensor Options

Sensor Menu options are described in the table below.

Table 4-37. Sensor Menu Options	
Option	Description
Action	Enables submenu for toggling status, updating, resetting, and clearing facts, and starting, stopping, and disabling services. Submenu will not be displayed on sensors that monitor multiple facts that use filters.
Armed	Disarms/Arms the policy. Action notifications are enabled when Policy is armed. Can be armed without stopping business view.
Transactions	Monitors transactions. Opens the TransactionWorks Explorer.
Display Fact	Opens the Deployment Tool and positions to the fact in the tree. If there is more than one fact, they are listed in a dropdown menu and the user can select one.
Acknowledged	Un-acknowledges/Acknowledges the selected sensor. Only enabled when sensors are started. When acknowledged, ACK is displayed immediately before the sensor name.

Table 4-37. Sensor Menu Options

Option	Description
Acknowledge...	Opens an <i>Input</i> screen where user can acknowledge a sensor with a note so that other users can view it.
Show Health	Displays, in percentage format, the health of the selected sensor.
Insert sub-view	Opens the <i>Load Sub-View From</i> screen for selection of Business View to be added. Only enabled when sensors are not running.
Save sub-view	Saves added sub-view to meshIQ Platform Core Services or your local machine.
New	Starts sensor wizard for defining a new sensor. Only enabled when sensors are not running.
New Folder	Adds a new folder to your selected sensor. Click Properties to define the new folder. Only enabled when sensors are not running.
New Link	Opens the <i>Link Sensor to Policy</i> screen to link a selected sensor to a specific existing sensor under a policy. The new link is added to your display. Only enabled when sensors are not running. Sensor links can be expanded within the Master View without having to open a new window.
New Clone	Opens the <i>Select Model Policy</i> screen where user can create a model sensor (template) to a previously saved business view. Ability to associate a sensor with a model sensor (template). Once the sensor is enabled it will load configuration options from the template. (Refer to Clone Sensor for details on this option.)
Copy	Copies parent sensor only. Copying sensors from a running business view to another running business view preserves the context/state of the running sensor.
Copy Recursive	Copies parent sensor and all child sensors.
Paste	Pastes the copied sensor to your selection. Only enabled after <i>Copy</i> or <i>Copy Recursive</i> is selected.
Apply Template	Contains default service definitions used during new service deployments written in .xml format. Each .xml template should contain only one service definition. Templates are only valid for domain and CEP server installations.
Delete	Permanently removes selection. Only enabled when sensors have been started.
Debug	Immediate debugging of selected sensor. Debug messages are located in the Business View log as configured in the policy logging section. This option is displayed by holding down the Shift key while right-clicking the started sensor.
Wizard	Displays the sensor wizard for redefining a sensor. Only enabled when sensors are not running. This option is not displayed on the parent sensor.

Table 4-37. Sensor Menu Options

Option	Description
Advanced	Ability to redefine the selected sensor without using the Sensor Wizard. This option is only displayed by holding down the Shift key while right-clicking the started sensor.
Properties	Opens the <i>Properties</i> screen to display and/or change your property settings. Only enabled when sensors are not running.

4.9.5.1 Clone Sensor

A clone sensor can be created that will take on the definition of a selected model policy. A model policy is a regular business view containing reusable rules. A clone sensor will morph into the business view it is cloning during runtime, as if that business view had been inserted in its place.


Model policies allow common business logic to be created and maintained in a single place. Changes made to a model policy will be reflected by all clone sensors that reference it in all business views upon restart. If you need to use the same rules in many different business views, you can create a model policy once and simply create a clone sensor where those rules should be used. For example: if you want to monitor five different servers, you can create one server model and then create four clone servers.

Model policies are located by default in `ds://model_policies` and can be changed by setting `property server.model.policy.folder=ds://<model policies folder>` in `global.properties` on domain server installation. Unlike link sensors, model sensors copy a model's definition at runtime when the policy is enabled. Changing a model sensor will affect all sensors that model after it only when the policy is restarted.

Table 4-38. Differences Between Clone Sensors and Link Sensors

Clone Sensor	Link Sensor
Copies a model's definition at runtime.	Links to an already deployed business view.
Model instance of business view.	Runtime instance of policy.

To **create a clone sensor**, do the following:

1. Save the business view to be cloned into the model policies folder.
2. In a different business view, right-click a folder sensor and select **New Clone** or click the *New Clone* link  to display the *Select Model Policy* screen.
3. Select the model business view to be cloned and click **OK**. The *Properties* dialog box is displayed. The only properties that can be changed are on the **Ignore** tab. This allows you to set a schedule different from the model (parent). If a field is not filled in, values from the model are used.

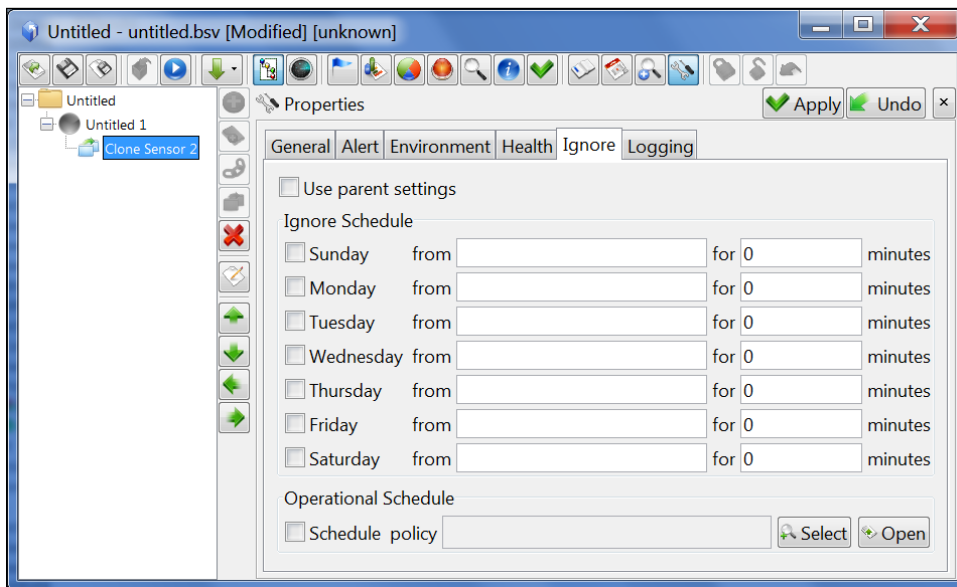



Figure 4-88A. Clone Sensor Properties

To **change an existing clone sensor** to clone a different model policy:

1. Right-click the clone sensor and select **Model From** to display the *Select Model Policy* screen.
2. Select the model business view to be cloned and click **OK**.

To **open the model business view** of a clone sensor, right-click the clone sensor and select **Open Model**.

4.9.6 Business Process

Business Process  is a tool supplied with the User Console that can be launched from the User Console's *Tools* menu. Business Process is used to create and view business views as a process flow rather than in a hierarchical form. These sensors are set up in the same way as business view sensors with the addition of a graphing capability on the properties screen.

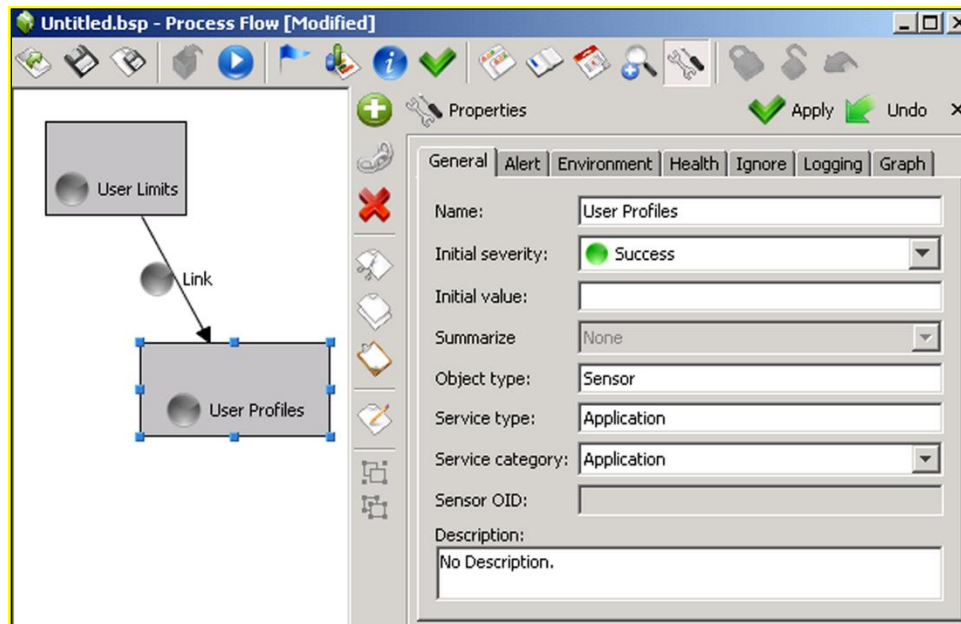


Figure 4-89. Process Flow Graphing Capability

To create a process flow, use the sensor wizard as you would for a business view. Each new sensor created will have its own rectangle with your designated sensor name inside. Connections are added by clicking one sensor on the blue dot and dragging your cursor to the next sensor's blue dot.

Configure or customize your process flow as follows:
















Table 4-39. Process Flow Configuration	
Icon	Description
Open 	Open a previously created business process.
Save 	Saves current business process as a .bsp file.
Save As 	Saves current business process with file type and name that you select.
New Sensor 	Creates a new sensor using the Sensor Wizard.
New Link Sensor 	Opens <i>Link Sensor to Business View</i> screen to link selected sensor to an existing business view. Sensor links can be expanded within Master View without having to open a new window.

Table 4-39. Process Flow Configuration

Icon	Description
Delete 	Deletes your sensor.
Cut 	Places selected sensor on clipboard. Sensor is removed.
Copy 	Copies selected sensor to your clipboard. Sensor is not removed.
Paste 	Pastes the cut or copied sensor to your selection.
Design 	Displays the first screen of the sensor wizard.
Group 	Groups selected sensors together. To select shapes, click each shape to be grouped while holding the Ctrl key. Click the <i>Group</i> icon.
Ungroup 	Ungroups sensors that have been previously grouped.
Apply 	Applies your last action.
Undo 	Undoes your last action.

To save your process flow, click the **Save** button in the toolbar. Your process flow will be saved as a .bsp file.


NOTE
Process flows can only be created or edited from the User Console.

4.9.7 Displaying Sensors Graphically

Business View and Business Process predefined sensor values can be displayed graphically for the selected root sensor, including all child sensors. The automatic sensor graph capabilities use charts and bar graphs to display sensor values. The graphs are configured based on which profiles you select. You may use the default properties or change them as you wish. The default properties as well as console profile properties are specified in each profile under `[AUTOPILOT_HOME]\naming\profiles\ <prof_name>`. All profiles are stored in Domain Server.

```

; Define AutoPilot console menu and toolbar.
;
; configure graphing options
property console.graph.refresh = 1000
property console.graph.usage.history.size = 5000

; graph dimensions
property console.graph.cols=3
property console.graph.bars=10
property console.graph.bar.ratio=7
property console.graph.min.bar.width=20
property console.graph.rows=10
    
```

```
property console.graph.max.charts=30
property console.graph.spacer=4

; font and stroke
property console.graph.font.name=Arial
property console.graph.font.size=12
property console.graph.stroke.size=3.0

; color coding
property console.graph.line.color.code=true
property console.graph.bar.color.code=true
property console.graph.label.color.code=true
property console.graph.fill=false

; gradient
property console.graph.fill.gradient=true
property console.graph.bar.gradient=true
property console.graph.line.gradient=true

; rendering hints (anti-aliasing)
property console.graph.render.hints=false


property console.view.default.map=false
```

Selecting a previously configured profile

1. Select **User > User Manager** from the toolbar.
2. Right-click profile to be used and select **Properties**.
3. Click the **Misc** tab.
4. Type name of profile (use same profile name as specified under `[AUTOPILOT_HOME]\naming\profiles\<prof_name>`) in the *Console Profile* field and click **OK**. (Refer to [Managing Users and Groups](#) for further information on managing users.)

Procedure to display sensors

To display sensors graphically, do the following:

1. Open the Business View or Business Process that you want to display sensors for.
2. Select the parent sensor that you want to view.
3. Click the graphs  icon. The graphs for parent and children will be displayed on the right of your screen. All display elements are based on the profile that you have selected.

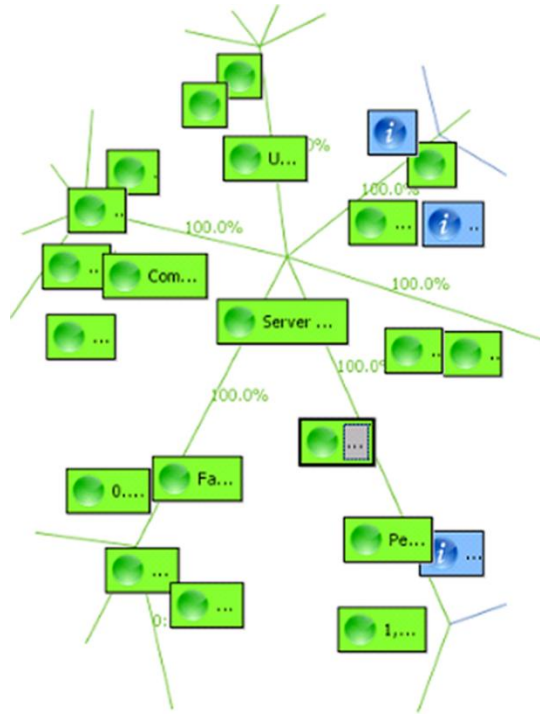


Figure 4-90. Sensor Value Graphical Representation-left side

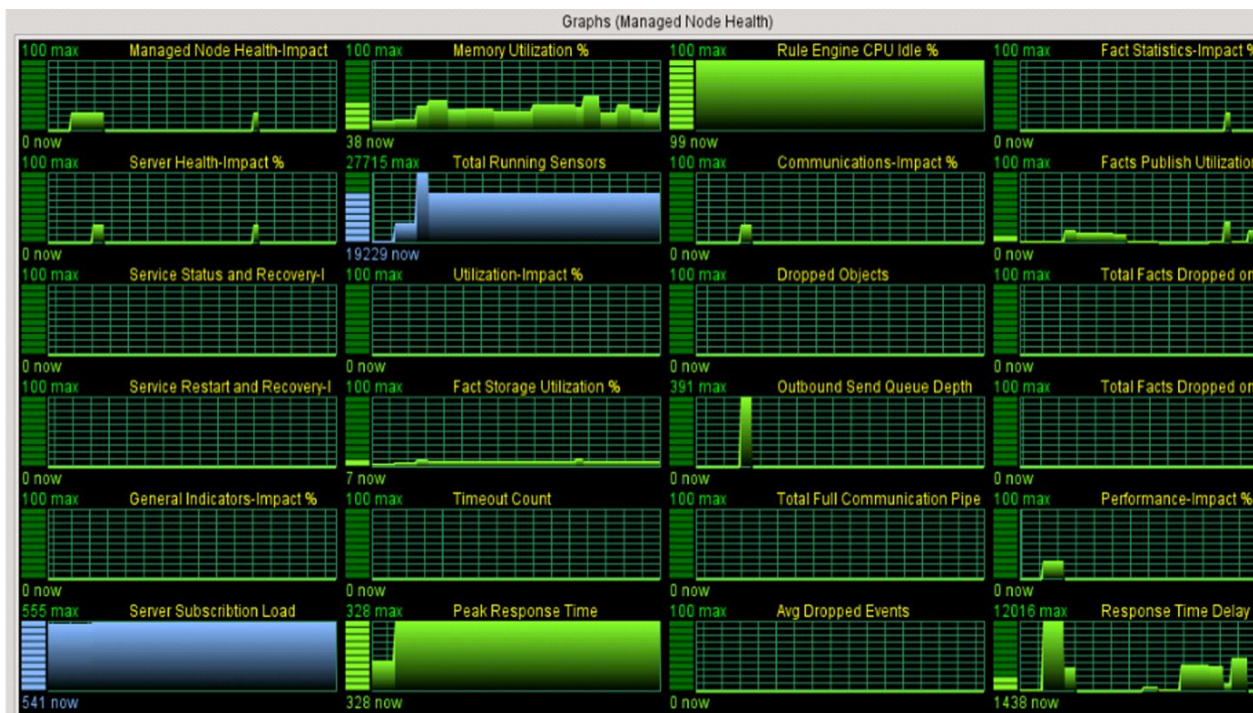


Figure 4-91. Sensor Value Graphical Representation-right side

4.9.8 Setting User Actions

The Action Manager allows users to set predefined user actions (scripts and executable programs) that can be executed in business views. There are three ways to set up an action.

- Through the Action Manager tool
- Through *Properties* screen under the *Alert* tab
- Through *Sensor Wizard* screens under the *Define State* screen.

These user actions must be in the path of the running CEP server or domain server where the business view is deployed, or the path must be specified within the *User Action* field of the Alert Details. Functions may include parameters that are passed on to the user action. Environment variables can be specified using `{env_var}` format, for example, `MYAPPL.BAT {SERVER} param1 param2...paramN`.

Within Action Manager or the action field of sensor wizard, native OS commands/scripts/executables can be executed on any CEP server (servers running Core Services CEP server) using the following:

`@${node}${user_action}` For example: `@$server1$start_command.sh`.

Defining an action using the Action Manager gives the user two distinct advantages. It allows the user to change an action without changing the business view. It also gives the user the opportunity to enable or disable a specific user action for all sensors at one time instead of enabling/disabling a specific user action for each sensor individually – as you would have to do using the *Sensor Wizard* or *Properties* screen. Refer to table and figure below for setting user actions.



IMPORTANT!

Parameters must be comma separated. Action invocation will fail if parameters have imbedded blanks.

Table 4-40. User Action Key Words/Prefixes

Key Word/Prefix	Description
Call	Java-based public static method within any classpath in the CEP server. For example: <code>call: com.nastel.nfc.util.GeneralUtils.getSeparator()</code>
Method *	Any method already used within Core Services with the format: <code>ServiceName.methodName(arg_list)</code> , where <code>ServiceName</code> is the name of registered service, such as an expert or manager. <code>methodName</code> is the name of the public method implemented by the service. <code>arg_list: type=value1,...type=valueN</code>

Table 4-40. User Action Key Words/Prefixes	
Key Word/Prefix	Description
	For example: method:JMX_Monitor.Invoke (String=Domain,String=MBeanName,String=mbean_method,String[]=)
@	Only used for executables and scripts. Runs action on system where the facts execute the action. The action is run from where the action is, not on the business view.

- * Supported method types can be String, Integer, Long, Float, Double, Boolean
Supported array types can be String[], Integer[], Long[], Float[], Double[], or Boolean[]

Array values are specified in the following format ['val1"val2'...'valN'] with no spaces between values. Example: String[]={This"is"a"test'} is a list of four strings. Empty values can be specified as null.

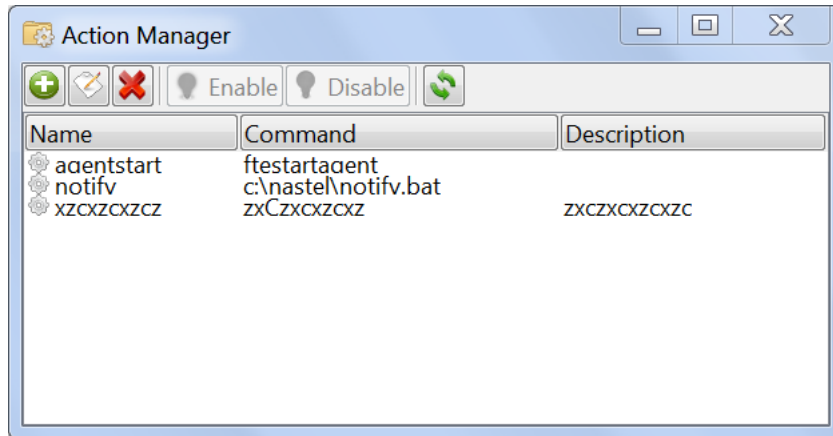




Figure 4-92. Action Manager Screen

To create a predefined user action, do the following:

1. Click the **Action Manager**  icon at top of the User Console screen. The *Action Manager* screen will be displayed.
2. Click the **New Action**  icon at top of this screen to create a new action. A window opens.

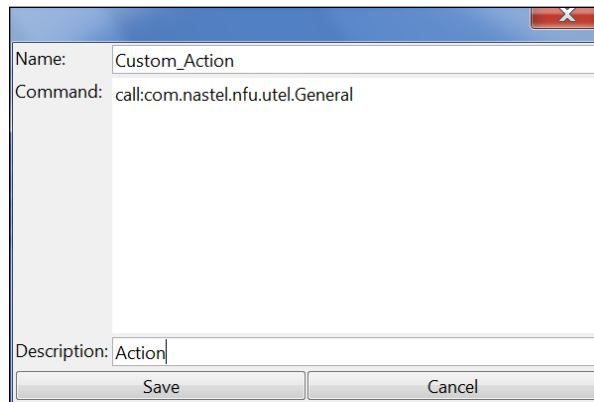






Figure 4-92A. Create Action Screen

3. Enter the name (logical) of the action.
4. Enter the command (physical) for the action.
5. Enter description of the action. (This is not required.)
6. Click **Save**. The action will be listed on the Action Manager screen (*Figure 4-92*).

Actions can be enabled  or disabled  as necessary by clicking the appropriate icon. When an action becomes disabled, it turns grey. To delete an action, click the delete  icon. Actions can also be enabled/disabled or deleted by right-clicking and selecting the action. To refresh your screen, click the refresh  icon.



NOTE

When actions are executed, they are retrieved from the Domain Server. With high volume actions, this can create high traffic between the executing CEP and the Domain Server. To cache the action, set the following environmental variable:

For Production – **property server.action.cache.active=true**

For Development – **property server.action.cache.active=false** (or undefined). This is the default.

4.9.9 State Change Delay

The *State change delay* field on the *Health* tab (Figure 4-93) designates the amount of time a new condition must be in effect before the state status displayed in the business views changes. This prevents the triggering of an alert when there is a spike in the sensor data. A value of 0 indicates no delay when the state changes.

The screenshot shows a configuration window with tabs: General, Alert, Environment, Health, Ignore, and Logging. The Health tab is active. Under 'Severity health', there are nine severity levels with their respective weights: Unknown (0.0), Emergency (0.1), Critical (0.2), Failure (0.4), Error (0.6), Warning (0.7), Success (1.0), Debug (1.0), and Information (1.0). Below this, there are four input fields: Default health (1.0), Weight (1.0), Auto-expire health limit (1.0), and State change delay (ms) (2000). The 'State change delay (ms)' field is circled in red.

Figure 4-93. Health Tab

The following is an example of the *State change delay* functionality.

In this example (Figure 4-94), the data is updated every 10 seconds and there is a 20-second delay. This means the condition must be true for 20 seconds for the state status to change in the business view. The state status is not necessarily updated every 20 seconds, but rather updated when the condition changes and remains in the new condition or below for 20 seconds. For simplicity, this example has only three conditions/statuses.



NOTE

On the upward severity, all timers between the current state and the new state start at the same time. For example, when the condition goes from green to red, both the yellow and the red timers start, implying the condition is “at least” yellow. If the condition dips down, the timers above the current condition stop, but the rest keep running. For example, if the condition dips from red to yellow, the red timer stops and the yellow timer continues to run. If the condition dips from red to green, the red and yellow timers stop, and the green timer starts. The condition must be true for the time specified in the *State change delay* field on the *Health* tab for the state to change.

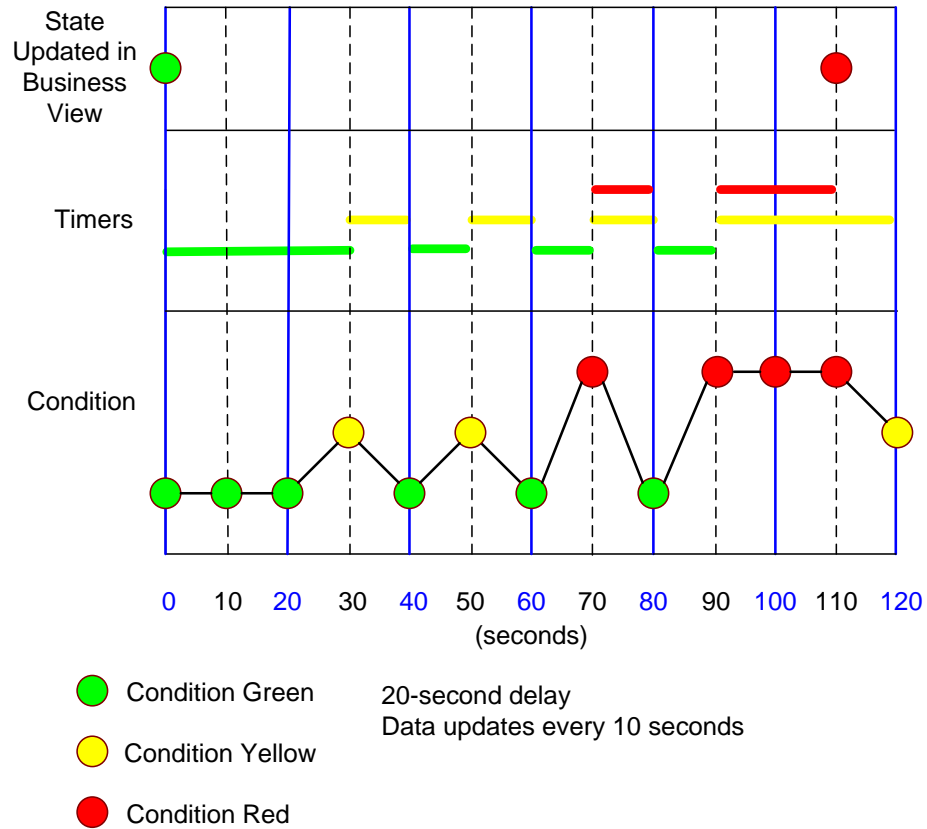



Figure 4-94. State Change Delay

The state starts as green. At 30 seconds it changes to yellow which starts the yellow timer. At 40 seconds the condition dips to green and the yellow timer stops. Since the system was in yellow for only 10 seconds, the state in the business view does not change.

The condition continues to fluctuate, but each change is only true for 10 seconds, so the state is not updated and remains green.

At 90 seconds the condition changes to red and stays red for 20 seconds causing the state in the business view to change to red at 110 seconds.



The *State Change Delay* functionality works in conjunction with the *Trigger delay* setting on the *Alert* tab. The settings on the *Alert* tab do not take effect until the delay has been reached.

4.9.10 Specifying Maintenance Schedule

There are two methods of specifying maintenance schedules in Core Services. Only one method (schedule) can be selected at a time.

- **Ignore Schedule** – selected sensors within a business view ignore faults and alert conditions during a specified window of time – at a specified time, any faults/alerts are ignored.
- **Operational Schedule** – a user-defined business view is set up to associate an operational schedule with a sensor(s) to control ignore states.

Ignore Scheduling

By specifying day and time periods for sensor maintenance window, alerts and user actions defined for normal operations can be blocked. Time periods set will be recurring until reset. If a sensor is set to ignore its monitoring on Sunday from 6 AM for 360 minutes (six hours, 6 AM to noon) the event that occurs during that time frame will be logged, but alerts and user actions will not be activated. The sensor severity status will not be propagated up the business view hierarchy and the sensor will have **IGN** displayed next to it.

Define maintenance window as follows:

1. Right-click sensor to be ignored.
2. In the sub-menu, select **Properties**, the *properties* screen will be displayed.
3. Click the **Ignore** tab to open *the Ignore properties* screen.

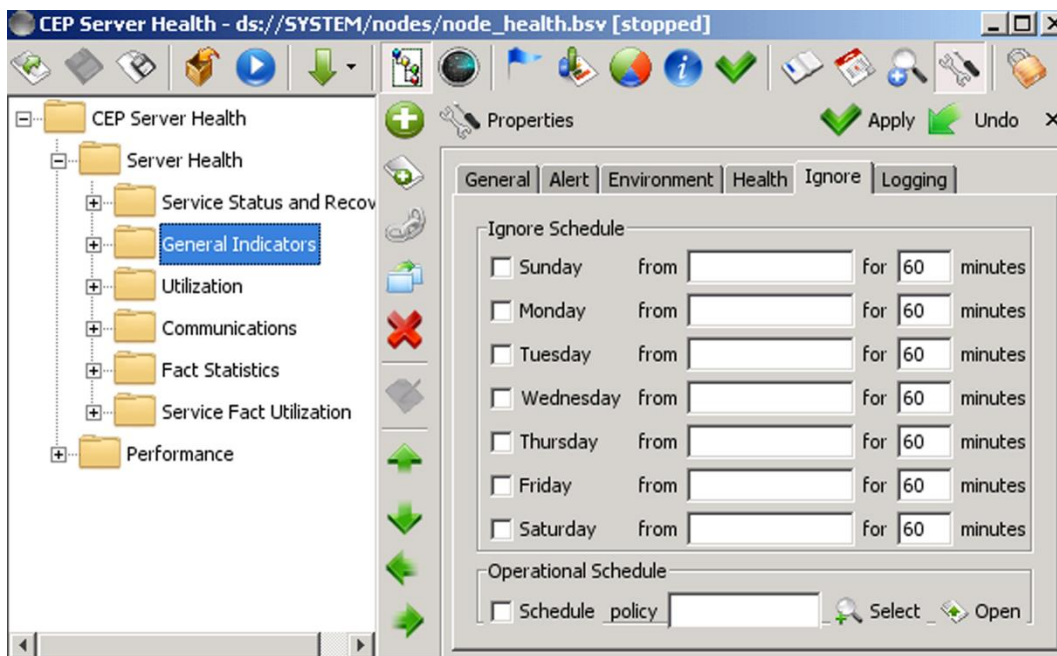


Figure 4-95. Opening Ignore Properties - default screen

4. For days to be ignored:
 - a. Enter time for the days you require.
 - b. Enter the number of minutes you want the sensor to be ignored. (Example: 12 Hours = 720 minutes, 24 hours = 1440 minutes.)

- c. Check (enable) the day. Sensor activity for the day and time specified will be ignored.
5. Ensure all other days are disabled (unchecked) to prevent inadvertent disruption of sensor data. When a day is disabled, the current time (default) for each day deselected is displayed, the day becomes grayed out and the ignore feature is disabled.
6. Click **Apply** when all day and time settings are set. All information for the specified period will be ignored.

The sample illustrated below shows a sensor set to be ignored from 5 PM, Friday until 8 AM, Monday.

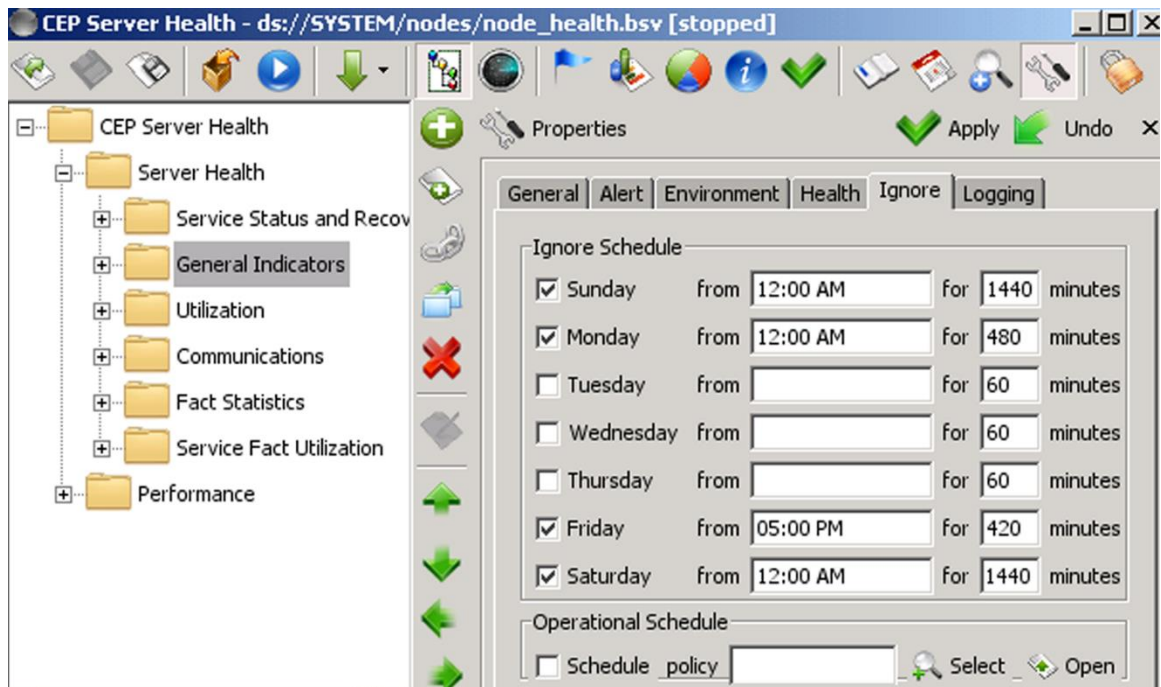


Figure 4-96. Maintenance Window -- Ignore Settings

Operational Scheduling

Users can define a rule-based operational schedule to select sensors to ignore faults and alert conditions as specified by a user-defined business view. This can be set up to change ignore settings as conditions change. If success (green icon) the sensor runs, if not success (icon any color other than green), the sensor is ignored. This provides more flexibility for the user to control his ignore settings.

The following is an example of a policy that can be used for an operational schedule:

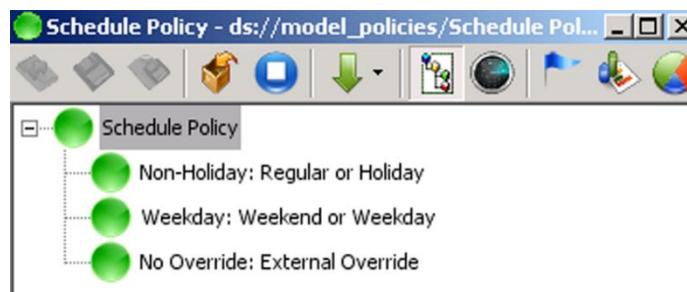


Figure 4-97. Example of a Schedule Policy

In this policy, a success (green) status is returned when not a holiday, weekday, and not overridden. The first two sensors are based on the date and day of the week. The Override is a value that can be sent with apfact to turn off the schedule at any time – such as a previously unplanned event. In addition to using apfact, various ways could be used to override, including the SQL query capability. Any policy can be used as an operational policy, and it does not need to be limited to date attributes. This policy was run at the Domain Manager for a central point of control but can be run at any CEP Server.

Set up an Operational Schedule as follows:

1. Create a customized policy to fit your operational schedule. (This policy will be selected in Step 5.)
2. Right-click the sensor to be ignored.
3. In the sub-menu, select **Properties**. The *properties* screen is displayed.
4. Click the **Ignore** tab to open *the Ignore* properties screen.
5. Click **Select** at bottom of screen. The *Select Policy as Operation Schedule* screen is displayed. Select policy and then click **OK**. (Click **Open** if you want to review your policy.)
6. Check (enable) the schedule.
7. Click **Apply**.

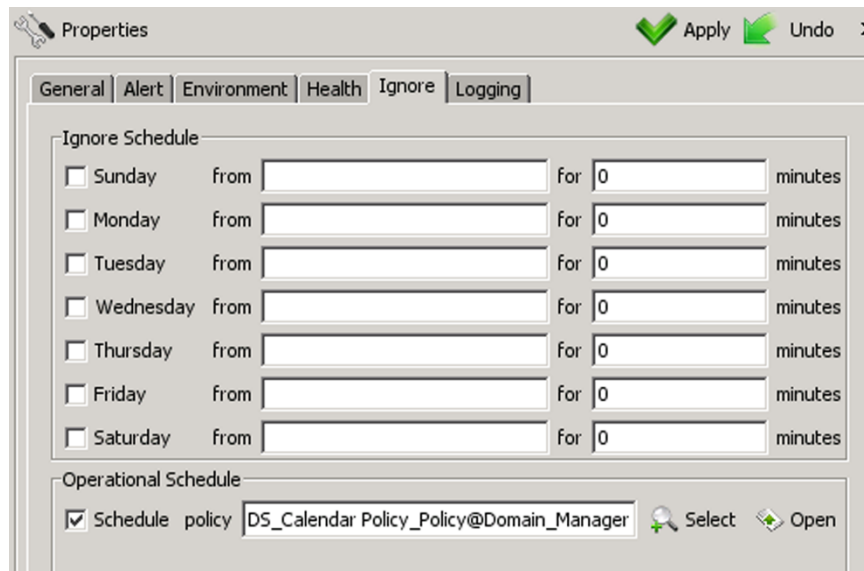


Figure 4-98. Maintenance Window – Operational Schedule

4.9.11 Ignoring Facts

A fact is ignored at the metric level. When a fact is ignored, actions and notifications (alerts) for all sensors that reference the fact are disabled. However, actions placed on the Sensor Evaluation page continue to be invoked even on an ignored fact.

To place a fact in an ignored state:

1. Right-click the selected fact.
2. Click **Action**. A sub-menu will be displayed.
3. Click **Toggle-Ignore-Status**.

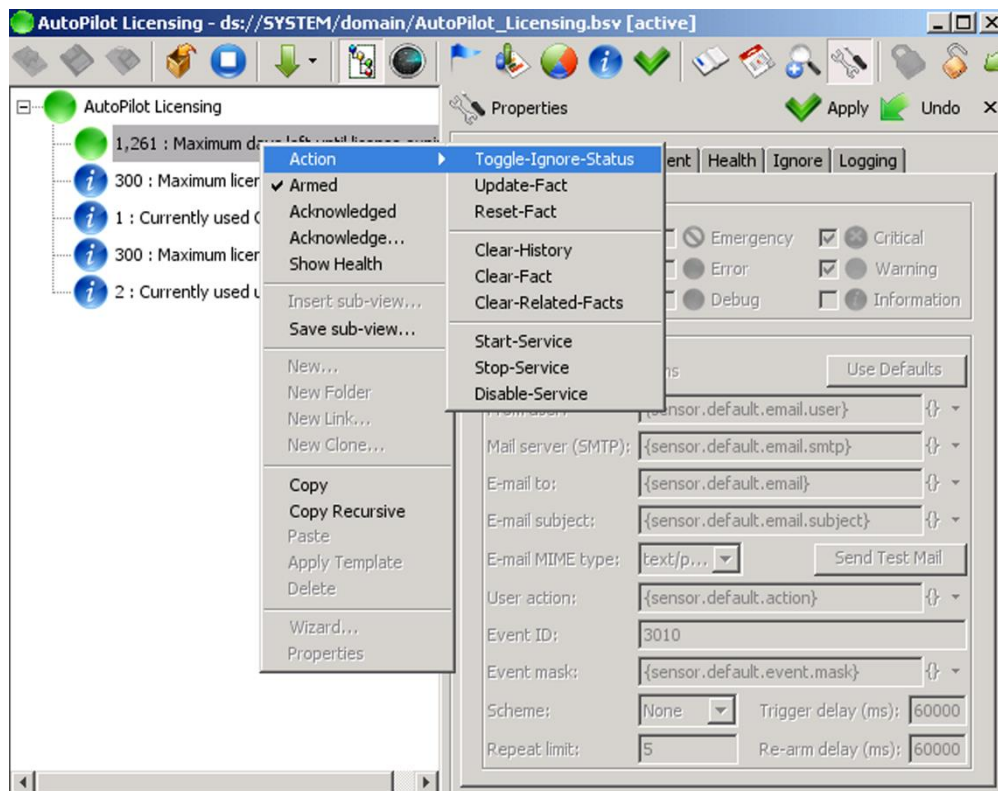


Figure 4-99. Toggle-Ignore-Status Menu

After the action is executed, **IGN** is displayed next to the selected fact. Every sensor that references this fact will stop sending actions and alerts. To reinstate the fact, repeat steps 1-3. **IGN** will no longer be displayed.

4.9.12 Maintaining Sensor History

Business views allow logging of sensor information to database and local flat files. This is useful when historical behavior of variables is required. Once logging is enabled, business view charts can be used to show sensor behavior over a period. It is recommended that key sensors record historical information. It will also help users understand trends and forecast future behavior.

The sensor logs are configured as follows:

Table 4-41. Log Sensor to Database

Property	Description
Exception based logging only	Click disable/enable button to log sensor information only when the sensor switches to any of the checked states under the <i>Alert</i> tab. When button is unchecked, sensor information is logged every time the value of the sensor changes.
Log sensor status to database	Click disable/enable button to log sensor status to selected database.
Use root sensor settings	If the sensor is a model, it will inherit the parent logging settings; that is, the database the parent logs to.
Database	<p>Select a database to receive the log from the menu:</p> <ul style="list-style-type: none"> • Oracle • SQL Server • Sybase • DB2 • Hypersonic SQL • ODBC Data Source • Informix • MySQL • Derby SQL <p>Note: JDBC drivers for the Derby SQL database must be installed separately.</p>
Database server	Enter the name of the database server.
Database name	User-defined file name.
Database table	The description is set when the format is selected.
Database user ID	The user ID is locally managed. See your DBA for local policy on DBA access and user ID. This user needs to be able to create a session, table, sequence, and triggers.
Database password	See your DBA for local policy on DBA access and password.
Test Connection button	Click Test Connection to verify your connection to your database. If the test is successful a <i>Connection Test Succeeded</i> screen will be displayed.
Create Table button	Click Create Table to create a new data table. May not work for some databases that do not support "number" as an SQL data type during table creation. Refer to Creating a Table Manually for more detailed information.
Use Defaults button	Click Use Defaults to use default properties that were defined in domain.properties on each domain server installation. Click Yes in Confirmation box.

When logging is configured, click **Apply**, in upper right corner, to save the logging properties. The business view will begin logging once the business view is deployed. The frequency of

recording depends on the frequency of change in sensor status, which is triggered either by changes in facts or changes in child sensors.

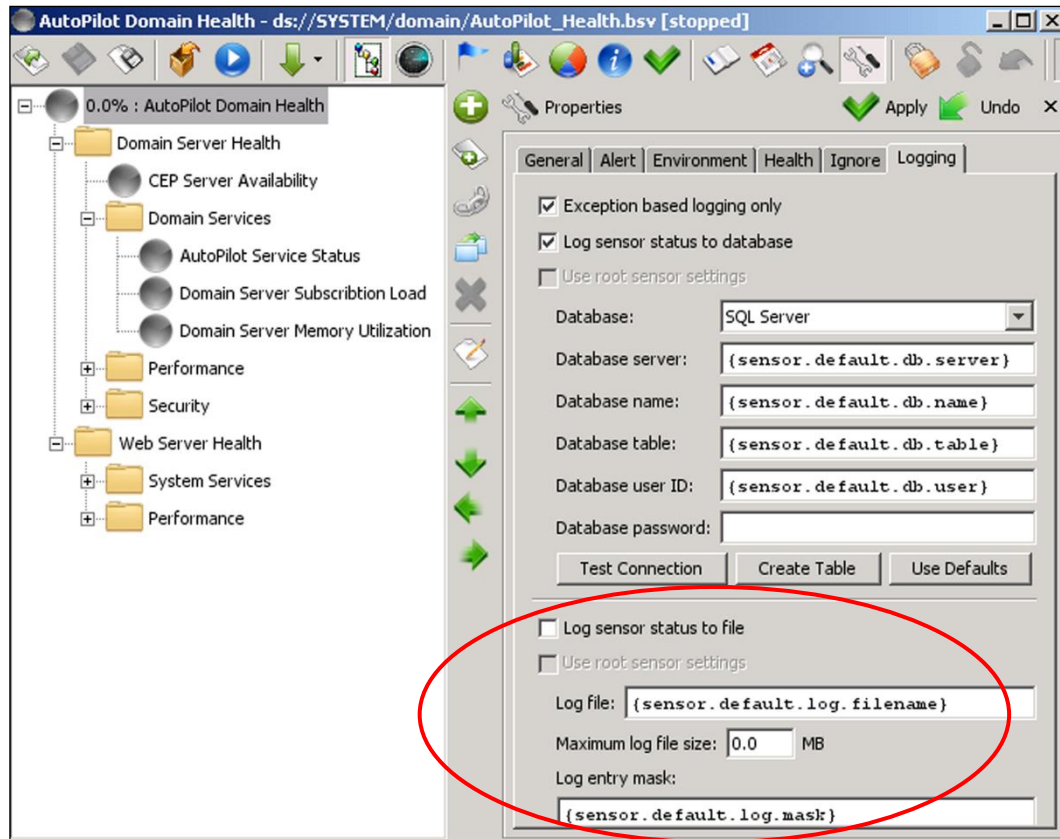


Figure 4-100. Sensor Logging Database Options

Table 4-42. Log Sensor Status to File

Property	Description
Log sensor status to file	Specifies to log sensor status to file. The default file format is EVT, the event log format.
Use root sensor settings	Specifies to use the root sensor settings as the default setting.
Log file	Enter a logical log file name with a .log extension and complete file address. (Example: <i>health.log</i>)
Maximum log file size	Define a maximum file size expressed in megabytes. (Example: 1.0 MB)
Log entry mask	Comma-separated list of user-defined tokens or variables such as %sevstr%, %f0, value, etc. in order to log additional entries.

4.9.12.1 Setting Up Pruning

Data can be summarized (daily) and purged from the sensor history tables. The process is called Aggregation. By default, sensor history aggregation is disabled (server.sensor.history.aggregate.retain.days=0). When enabled, it is kicked off every day at

00:01:00, where data older than five days is summarized and purged from sensor history tables. Aggregation is kicked off on every start and then each subsequent day at 00:01:00.

Scheduling parameters can be changed by defining/modifying the following properties:

```
// controlling the initial aggregation task (initiated on start
only)
server.sensor.history.aggregate.schedule.first=10 (minutes)
server.sensor.history.aggregate.schedule.first.delay=30000 (ms)

// controlling fixed-rate scheduled aggregation task
server.sensor.history.aggregate.schedule.hour=0
server.sensor.history.aggregate.schedule.minute=1
server.sensor.history.aggregate.schedule.second=0
server.sensor.history.aggregate.period=86400 (seconds in 1 day)
server.sensor.history.aggregate.retain.days=2 (retain 2 days'
worth of history) setting to 0 disables aggregation altogether.
```

Additional aggregation metrics have been added under:

```
<Server>_Facts\Logging\Aggregation
```

The aggregation automatically determines the tables that need to be summarized by querying "db_services" table. All summary data is recorded into db_sensor_summary table, which must be created first as follows:

MySQL:

```
CREATE TABLE `m6_sensors`.`db_sensor_summary` (
  `SID` int(11) NOT NULL,
  `SensorName` varchar(255) NOT NULL,
  `SensorTable` varchar(255) NOT NULL,
  `SensorDate` timestamp NOT NULL,
  `Records` int(11) NOT NULL,
  `Average` float NULL,
  `Maximum` float NULL,
  `Minimum` float NULL,
  `Deviation` float NULL,
  PRIMARY KEY (`SensorName`,`SensorTable`,`SensorDate`,`SID`)
);
```

SQL Server:

```
CREATE TABLE [dbo].[db_sensor_summary] (
  [SID] [int] NOT NULL,
  [SensorName] [nvarchar] (255) NOT NULL,
  [SensorTable] [nvarchar] (255) NOT NULL,
  [SensorDate] [datetime] NOT NULL,
```

```

[Records] [int] NOT NULL,
[Average] [float] NULL,
[Maximum] [float] NULL,
[Minimum] [float] NULL,
[Deviation] [float] NULL,
PRIMARY KEY (SensorName, SensorTable, SensorDate, SID)
);

```

4.9.12.2 Creating a Table Manually

Refer to [Appendix F](#) tables to develop your table headings for manual table creation while logging sensors to a database.

The Service Category Table is defined below.

Table 4-43. Service Category	
Service Category Number	Service Category Name
0	Hardware
1	Network
2	Server
3	Operating System
4	Middleware
5	Database
6	Application Server
7	Web Server
8	Web Service
9	Client
10	Application
11	IT Service
12	Business Service
13	Transaction
14	Policy
15	Miscellaneous
16	Other

4.9.13 Sensor Performance Counters

Sensor Performance Counters are published under the `[managed_node]_Facts/Facts/Sensor` category. All counters are computed since the last reset and must be stopped and started to be reset.

Table 4-44. Performance Category

Category	Description
<code>absolute_rate_rules_per_sec</code>	number of absolute rate rules per second
<code>absolute_rate_sensors_per_sec</code>	number of absolute rate sensors per second
<code>last_sensor_exec_time_ms</code>	time in ms. taken by the last sensor execution
<code>max_sensor_exec_time_ms</code>	maximum sensor execution time in ms
<code>min_sensor_exec_time_ms</code>	minimum sensor execution time in ms
<code>total_processed_rules</code>	total number of processed rules since last reset
<code>total_processed_sensors</code>	total number of processed sensors since last reset. (Sensors may execute one or more rules)
<code>rate_rules_per_sec</code>	processing rate of rules per second
<code>rate_sensors_per_sec</code>	processing rate of sensors per second
<code>sensor_idle_percent</code>	percent of time spent outside of sensor processing
<code>sensor_busy_percent</code>	percent of time spent processing sensors and rules
<code>total_sensor_time_ms</code>	total time in ms. spent processing all sensors
<code>total_processed_rules</code>	total number of rules processed
<code>total_processed_sensors</code>	total number of processed sensors
<code>sensor_turn_around_time_ms</code>	measures the time it takes to deliver facts to sensors

Table 4-45. Runtime Category

Category	Description
<code>sensor_arrival_rate_per_sec</code>	total number of sensors arriving per second
<code>sensor_delivery_rate_per_sec</code>	total number of sensors being delivered per second
<code>sensor_total_actions</code>	total number of executed actions (sensor user actions)
<code>sensor_failed_actions</code>	total number of failed actions (sensor user actions)
<code>sensor_total_notifications</code>	total number of successfully sent SMTP messages
<code>sensor_failed_notifications</code>	total number of failed notifications
<code>sensor_rule_backlog</code>	total number of sensors backlogged
<code>sensor_total_expiring_threads</code>	total number of expiring threads

sensor_total_action_threads	total number of outstanding action threads
sensor_total_running	total number of sensors running

4.9.14 Documenting Sensor Information

Sensor information including facts, logic, conditions, alerts, etc. can be saved/exported in the following formats:

- HTML
- PDF
- RTF
- XML

1. Click **Business View Explorer** from the **Tools** menu bar.
2. Expand the folder that contains the Business View to be saved.
3. Right-click the business view and select *Create Sensor Documentation* and then the desired format.

This is especially useful for viewing by other users that are not owners of the business view.

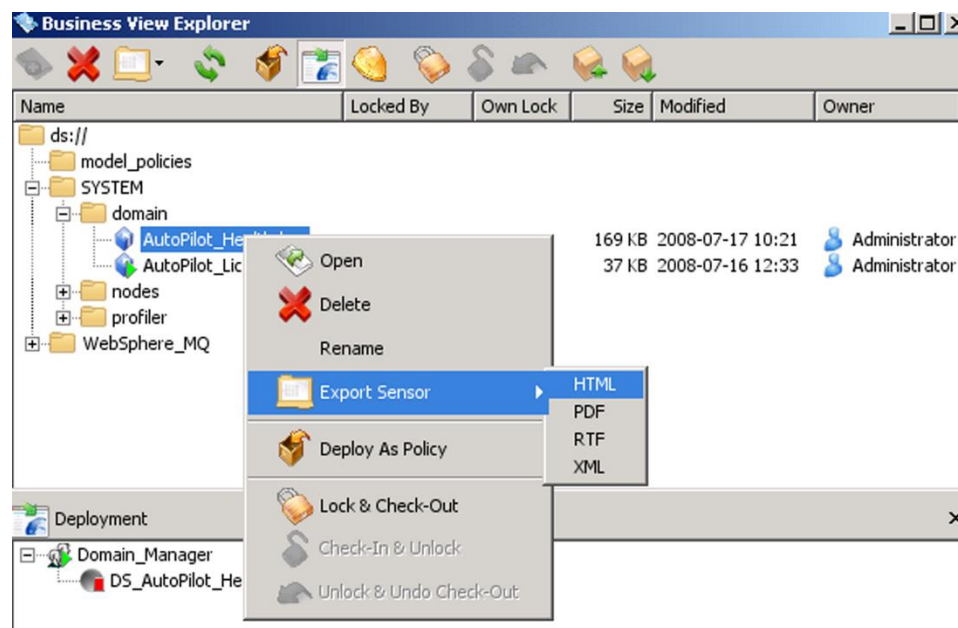


Figure 4-101. Creating Sensor Documentation

4.10 Real-Time Monitoring

This section will discuss the real-time monitoring of business processes, events, and performance issues. meshIQ Platform Core Services has two distinct real-time monitoring capabilities: monitoring of business views and performance monitors. Real-time monitoring enables users to proactively monitor system-wide performance, diagnose system and application bottlenecks, efficiency, congestion, and latency using real time and historical statistics.

Monitoring your environment will affect system performance; the degree of performance degradation depends on user-defined sampling rates, the size of the environment, and the number of monitored resources.

4.10.1 Monitoring Business Views

Business views are a collection of rules that define a desired state of the environment. Your deployed business views give you real-time views of the information in the form you defined.

Monitoring Business view lets you see the status of applications and identify the impact of failures on the overall environment, enabling you to take prompt actions in response to the events as they happen.

There are many options available when viewing business views. Based on your needs, when monitoring business views, you can select from several options that offer a good deal of information. You can monitor any single view, or any combination of views of any number of business views. By opening multiple instances of the same business view, you can view multiple detailed areas simultaneously. Open any business view by right-clicking it and then selecting *Open* in the sub-menu.

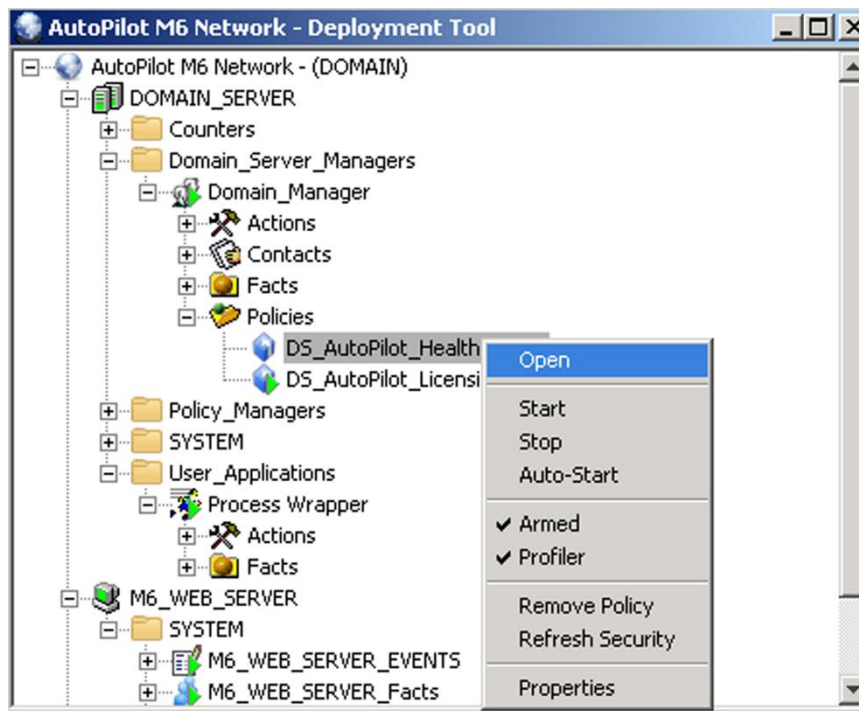


Figure 4-102. Open Any Business View

Typically, the business view will display the configuration from when it was last viewed. In the sample below the *Severity*, *Health* and *State* are all displayed. Business views offer a

variety of monitoring options. By selecting options from the toolbar, you can open and close any required monitoring screens.

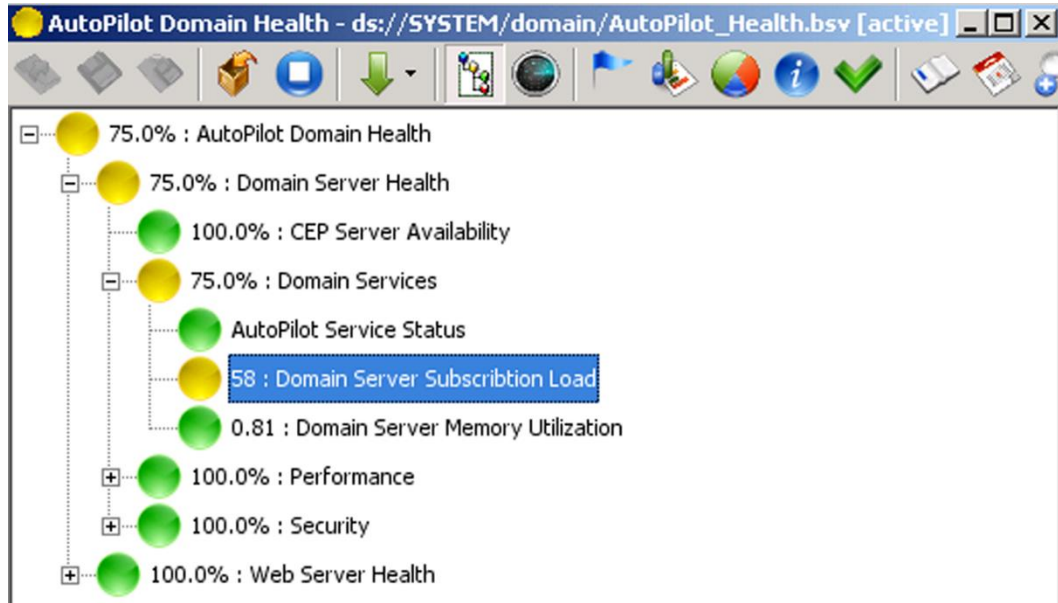



Figure 4-103. Monitoring a Working Business View

4.10.1.1 Event Logs

Event views in business views work exactly the same way as in the console. Open the *Event Viewer* to access the detailed information available. The events files you select will be displayed. It can be the file with events related to the business view displayed, or any other file.

Table 4-46. Business View Event Logs

Property	Description
Severity	Depicted with the relative icon that represents the current health
Date/Time	Displays the date and time of the log entry
Source	Identifies the source of the entry as applicable
Event ID	Provide available event numbers
Account	User account ID for owner of event sources
Message	Contain descriptive information about the event. Contents will vary with the nature of the sensor.
Event Detail	Provide detailed information about the event, source, and status. To open the detail window double click the event you want to view. You can copy the detail by clicking the Copy All  icon at the bottom right of the screen.

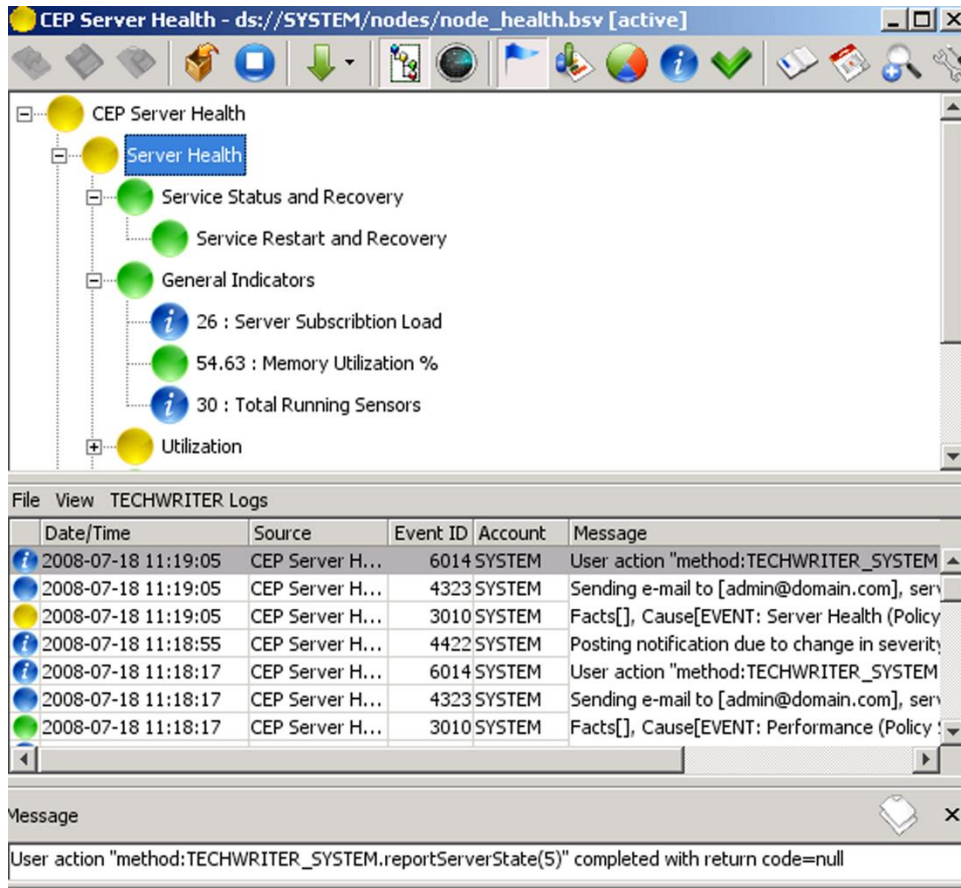


Figure 4-104. Business View Event Log

4.10.1.2 Charting Business View Metrics

The charting option available in business views is different than those available in the console charting tool *Open Charting*. Value charting is the default.

A Business View may be running while adding sensors into the charting index.

NOTE

Drag and drop sensors into the charting index or click the **Add** button. The sensor will be assigned a color for charting.

Table 4-47. Business View Charting Tool	
Property	Description
Source	Real Time: Plots from current sensor status in real time.
	History: Plots from local log file archives. If the logging option is not active in the expert or manager, there will be no history available.
Data	Value: Plots based on the numeric value of the facts monitored.
	Severity: Graphic will display to reflect the severity status of the sensor for the time defined.

Table 4-47. Business View Charting Tool

Property	Description
	Health: Plot will reflect the health of the business view for the prescribed time frames.
Zoom	Top: Move to the right to scroll across the chart from left to right.
	Bottom: Use to expand or enlarge the charted area for a given time to give greater detail.
Type	There are seven charting formats available: <i>Step, Scatter, Bar, Mixed, Line, Area, and Area-Mixed</i> . Click the format that best suits your needs. Charting formats can be changed at any time without interrupting the charting process.
Add	Click to add selected sensor to the charting index.
Update	Click to update the chart in real time.
Period	The time intervals at the bottom of the draft represent the timeline of the chart being viewed. Time can be measured in minutes, hours, days, or weeks. If the chart is from archived files, it will display the times as relative to the file, not the current time.
Legend	Correlates color to actual sensor being monitored.
Sensor	Displays sensor state, value, and name.
Value	Displays current sensor value rounded to the nearest hundredth.
Min	Displays the minimum value since charting was started.
Max	Displays the maximum value since charting was started.
Count	Displays how many samples were taken since charting was started.
Average	Displays the average value since charting was started.

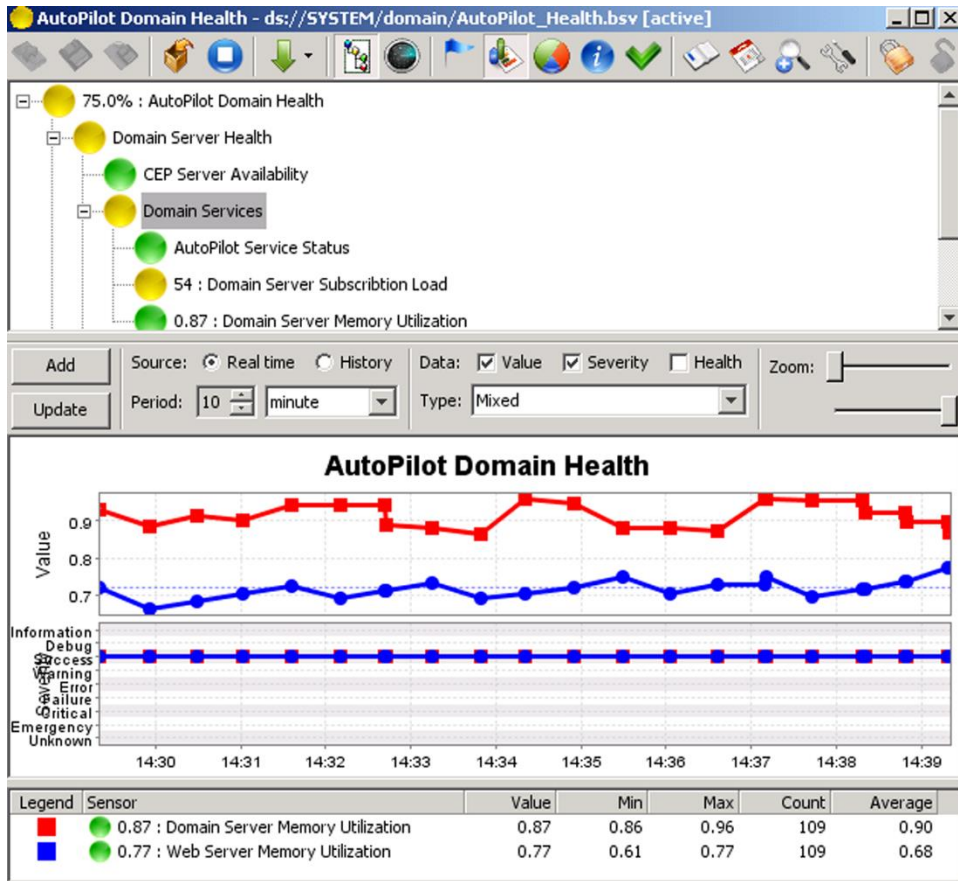


Figure 4-105. Business View Charting Options

Plotting Severity

All the sensors being charted with the same status will be represented by a single line on the chart. The look varies with the type of chart selected.

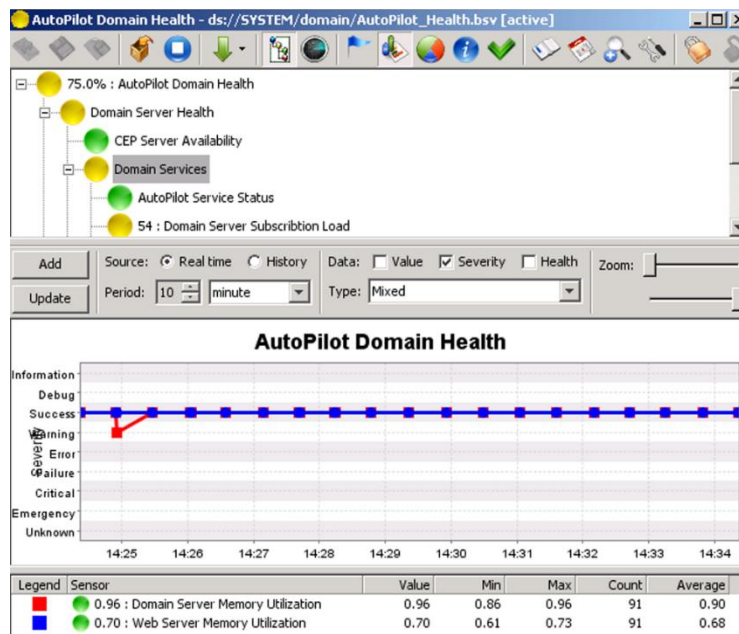


Figure 4-106. Severity Chart Format Samples

Plotting Health

As with severity, the common health levels are grouped, the chart reflects the percentage of health. The percentages are defined in the health parameter in the sensor properties.

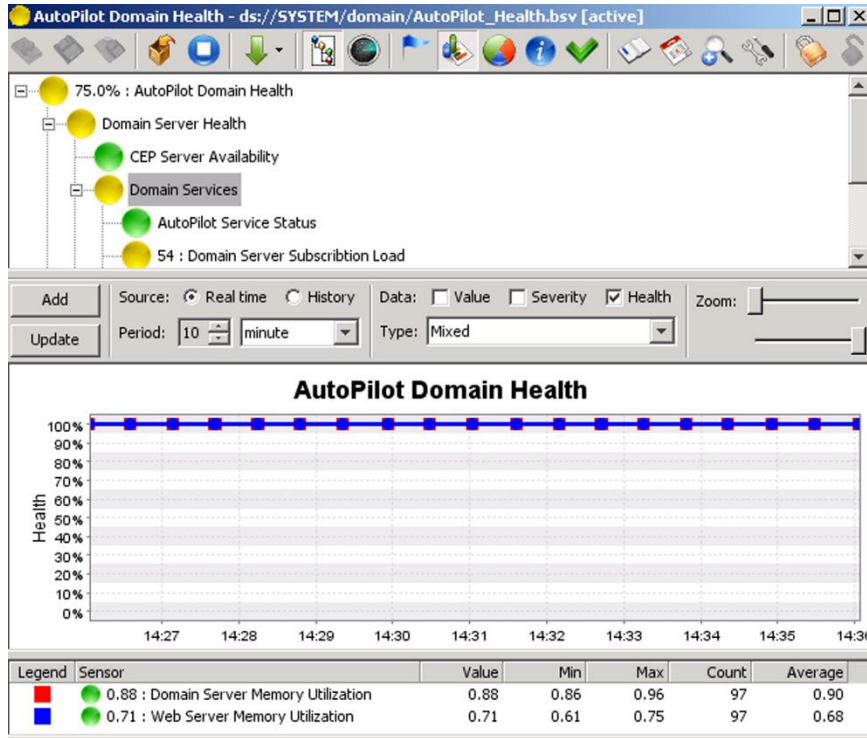


Figure 4-107. Health Chart Format Samples

Plotting Value

Charting with value is only effective with numeric sensor results. As you can see in the sensor index the Memory Utilization sensors are prefixed with a numeric value. Each sensor's value is plotted in the chart as the chart refreshes.

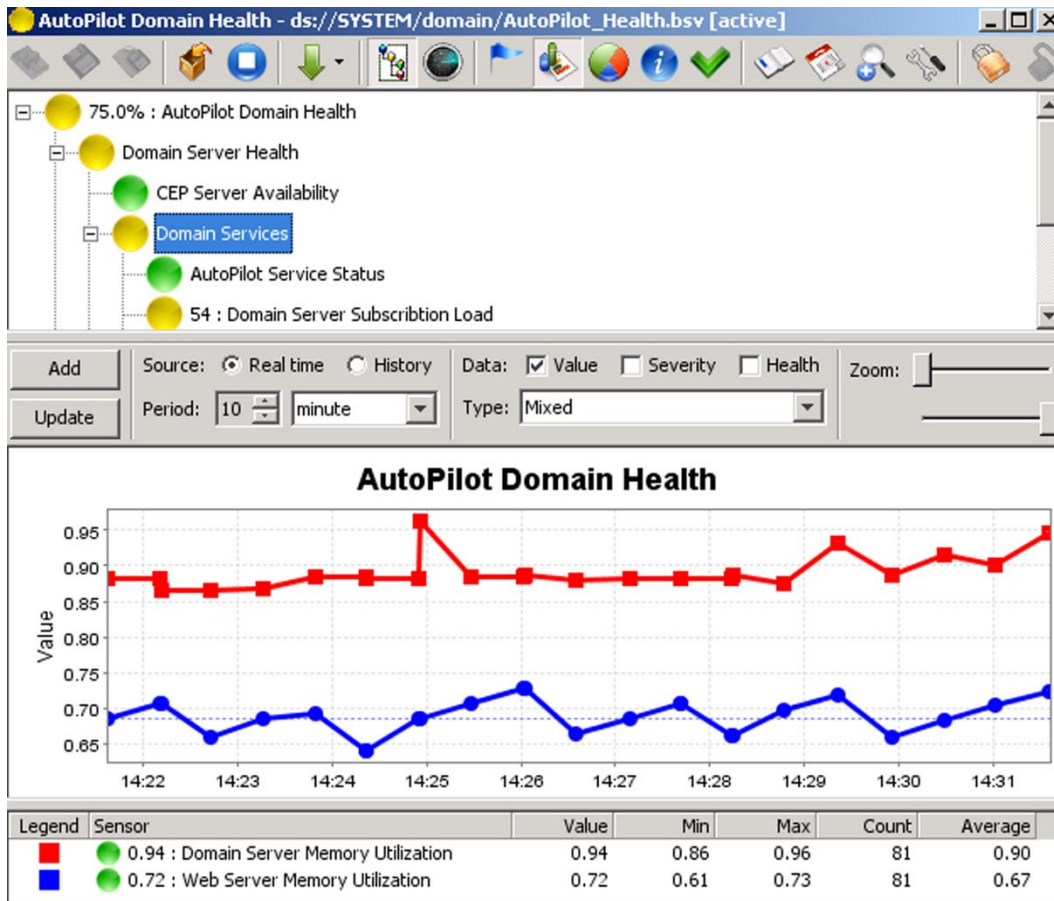


Figure 4-108. Value Chart Format Samples

4.10.1.3 Chart Options

Right-click in the charting area of the screen to access the sub-menu. There are several additional options available to you there.

Table 4-48. Chart Options	
Option	Description
Properties	Open the chart properties. The chart properties allow you to customize the look of your charts and alter the ranging options.
Save As	Captures the chart graphic and saves it to a user designated file.
Print	Prints the chart and legend to local printer.
Zoom In/Out	The zoom enables you to view the chart in greater or less detail.
Auto Range	Allows you to range in or out on both axes together or either horizontally or vertically, independently.

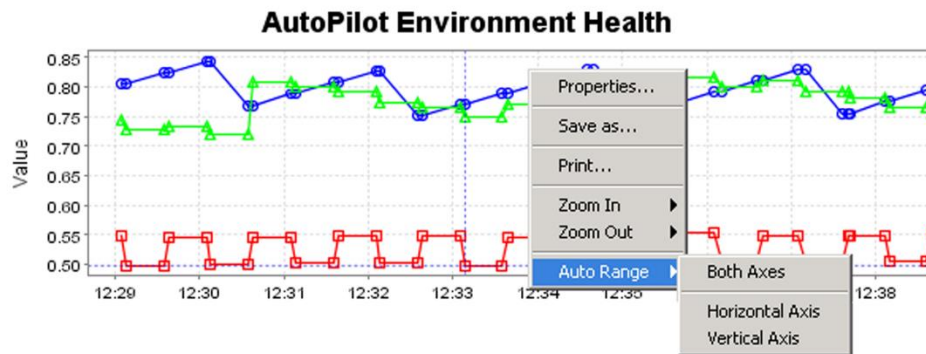


Figure 4-109. Chart Options

Hovering the cursor over any data point in the scattered, mixed, or area mixed chart mode will open a flag that displays relevant data and status. Clicking on that point will set crosshairs and allow you to zoom in or out from that spot.

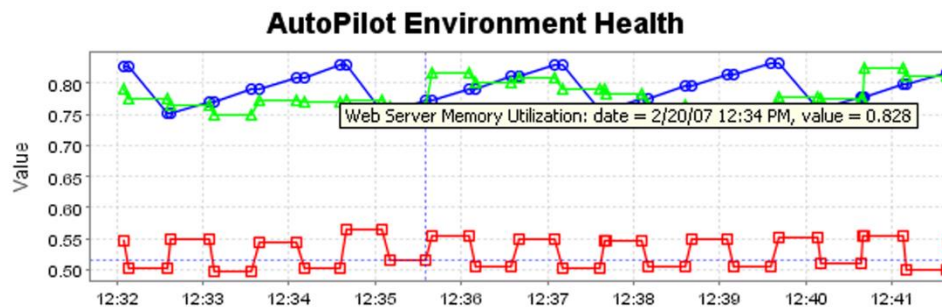


Figure 4-110. Charting Options

4.10.1.4 Sensor Overview

The Sensor Overview provides a summarized look at the business view configuration. It identifies key settings within the sensors that were defined when the sensor was developed or modified.

Table 4-49. Business View Sensor Overview	
Sensor	The identity and status of the sensor
Alert	Identifies if there are alerts specified in the sensor
Environment	Identifies if environmental variables are applied to the sensor
Ignore	Identifies if the sensor is to be ignored during any period
Logging	Identifies if the events are being logged.

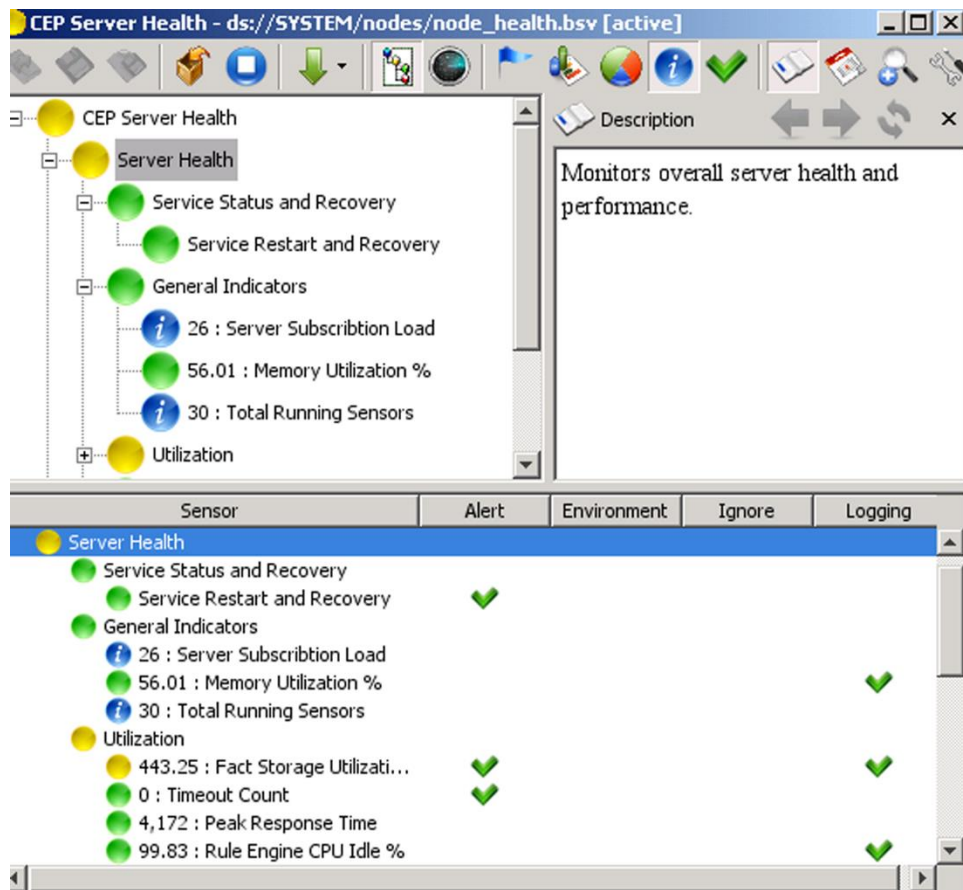



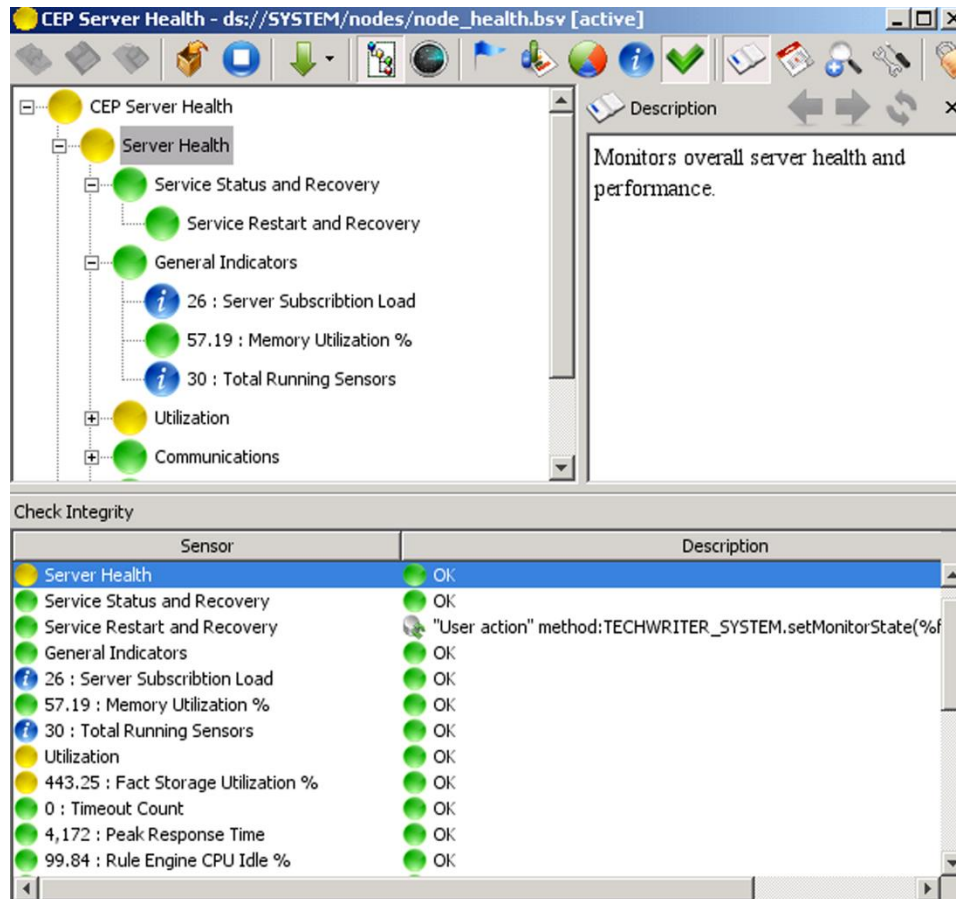
Figure 4-111. Business View Sensor Overview

4.10.1.5 Checking Sensor Integrity

Click the Integrity  button to view the status of the business view and all of its sensors. This checks for common user errors associated with each sensor. The following conditions are checked:

- Sensor description is missing
- `[Field Name]` contains undefined environment variable(s)
- Fact `[#]` refers to an undefined service
- Fact `[#]` refers to an unavailable service
- Fact `[#]` included but not being used
- Fact `[#]` is undefined


If a sensor does not have an error, the description column displays a Success alert. If a sensor has an error, the description column displays a warning alert and the reason for the warning. Click **Check Integrity** to refresh the screen after making changes to the sensor.



Sensor	Description
Server Health	OK
Service Status and Recovery	OK
Service Restart and Recovery	"User action" method:TECHWRITER_SYSTEM.setMonitorState(%f
General Indicators	OK
26 : Server Subscription Load	OK
57.19 : Memory Utilization %	OK
30 : Total Running Sensors	OK
Utilization	OK
443.25 : Fact Storage Utilization %	OK
0 : Timeout Count	OK
4,172 : Peak Response Time	OK
99.84 : Rule Engine CPU Idle %	OK

Figure 4-112. Business View Sensor Integrity

4.10.1.6 Sorting Temporary Dynamic Sensors

Click the **Sort Dynamic Sensors**  button to sort temporary dynamic sensors by Name, Severity, Value, Health, and in Ascending Order and Descending Order. Only sensors selected as *temporary* from the sensor wizard can be sorted.

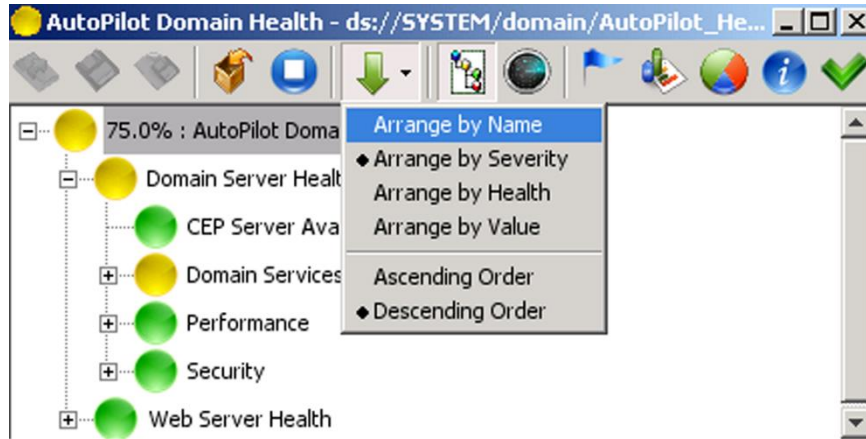



Figure 4-113. Arranging Dynamic Temporary Sensors

4.10.1.7 Showing Description

Display the **Description**  to view the functional description of the sensor or business view. The developer generally provides the description. The description displayed is of the selected or highlighted item. By scrolling through sensors each unique description will be displayed.

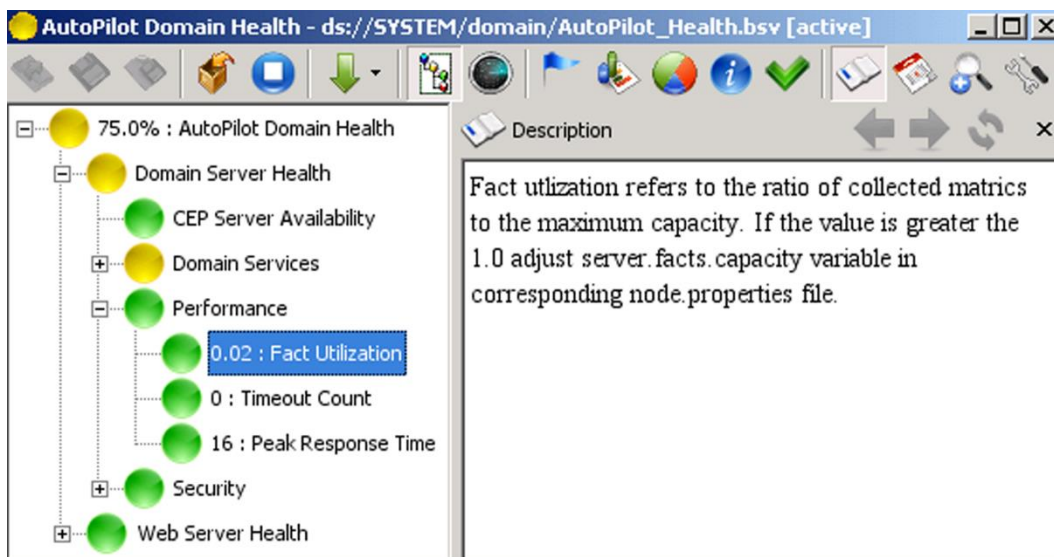



Figure 4-114. Business View Sensor Description

4.10.1.8 Showing Facts

Displays the **Facts**  used in a given sensor. If the business view is currently active the facts displayed will reflect the real-time status. Real-time facts view is limited to 1024 display entries to offset display problems associated with too many entries. The facts displayed when the sensors are inactive reflect the last active state.

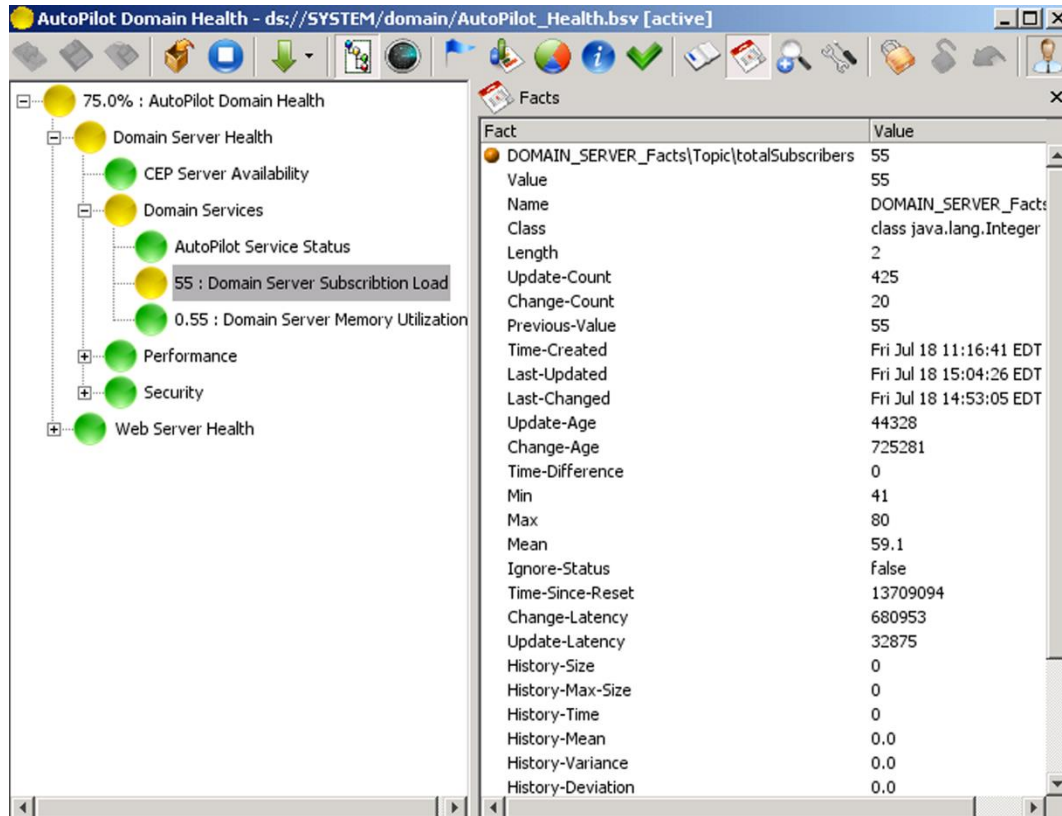



Figure 4-115. Business View Sensor Facts

4.10.1.9 Related Views

Open **Related Views**  button to show all active business views that are related to the current view. Related views are those that monitor similar sets of metrics. meshIQ Platform servers perform this correlation automatically.

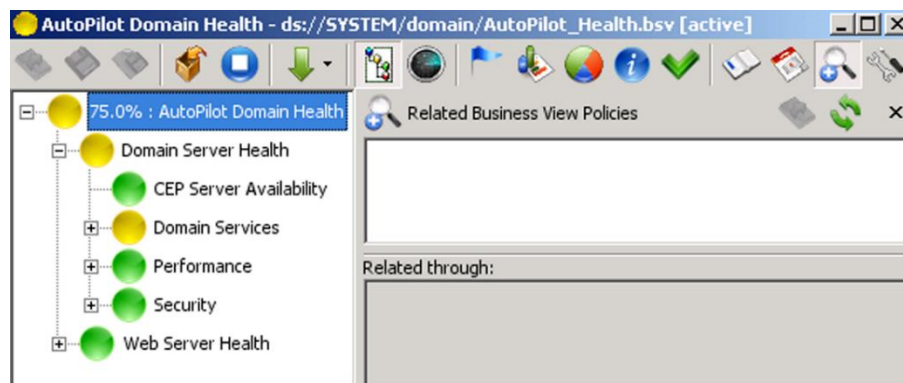


Figure 4-116. Related Views

4.10.1.10 Business View Policy Properties

Display the **Properties** of the business view/policy to review or change when needed by right-clicking on the business view/policy, then click **Properties**. The Business view file is specified as: `ds://folder/file_name.bsv`, where `ds://` is a URL extension for the meshIQ Domain Server. Use Business View explorer tool to create folders, save business views and deploy them to the appropriate manager(s). It is recommended that all business views be stored at the domain server and deployed from the domain server out to the CEP servers. This reduces business view maintenance, improves performance and productivity. It also eliminates business view file synchronization issues.

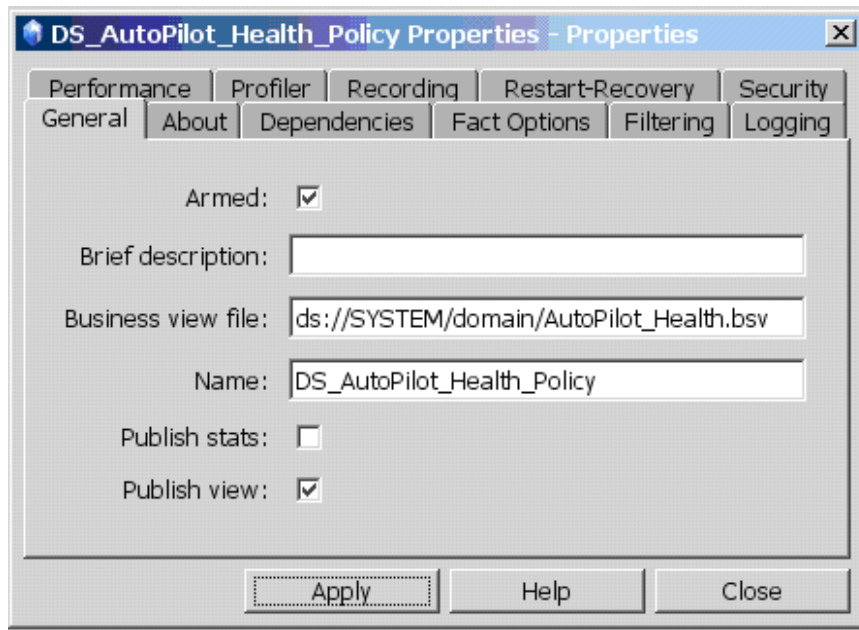


Figure 4-117. Business View Properties

4.11 Event Logging

Event logs contain data about the operation and status of your system. Business views record information into event logs, on every alert or action that is taken on behalf of a user or user-defined rules.

The logging data is based on the logging characteristics defined by the user while configuring experts, managers, policies, and business views. Event logs track system activities such as the start and completion of jobs, device status and health, system events, alerts, and notifications.

The collected system data is compiled and recorded in user-defined generated event logs. Event logs help you monitor and control system activity, analyze and correct problems. meshIQ Platform Core Services installations maintain a set of logs for all installed components in `[AUTOPILOT_HOME]\logs` directory. For more information, refer to [Troubleshooting Techniques](#).

The event logs have an `.EVT` file extension (example: `DOMAIN_SERVER_SYSTEM$0.EVT`).

Event logs have the *Logical* and *Physical Name*:

- **Logical Name:** User known name that is displayed in the event viewer title bar and in individual service Logging Properties. Default Logical Names are *System Unnamed*, and *Services*. There are system logs and user-defined logs.
- **Physical Log Name:** Actual system stored file name. Each log file has a physical file name; the actual file is stored on the file system. The physical name convention is: `[MANAGED_NODE_NAME]_[LOGICAL_NAME]$0|1.EVT`
(Example: `DOMEGAX_SYSTEM$0.EVT`). Since logs are circular, Core Services may allocate two logs, for example: `DOMEGAX_SYSTEM$0.EVT`, when log is full, the `$0` log is moved to `DOMEGAX_SYSTEM$1.EVT` and the original file is truncated (deleted and recreated).


System Logs

- **SYSTEM:** System services record to this log.
- **SECURITY_ACTIVITY:** (Domain Server only) security service records information on security action. The log is created only when "service activity" is enabled on the `DOMAIN_SERVER_SECURITY` service.
- **ACTION_ACTIVITY:** Contains failed user actions
- **UNNAMED:** Should always be empty. This log is maintained when an event is written to a non-existent log. Any events recorded into this log indicate an internal error and should be reported to [meshIQ technical support](#).

User-defined Logs

- **SERVICES:** Where all management services record failures, errors, and warnings as well as trace messages.

- **User-defined:** The log name can be changed/created when you modify individual service's *Logging Properties*.



All log files are stored as plain text files, viewable using any text editor.

NOTE

The events in event logs are formatted as follows:

- **key:**{time=*timestamp*}{type=5}{eid=*id*}{account=*user*}{from=*component*}{event=*message*}.
- The log is formatted to provide all the necessary data for the event displayed. Individual fields can be expanded to display the full detail, or you can use the detail window to display all event details from all fields.

Table 4-50. Event Log Properties	
Field	Description
Severity	The related icon for the severity state or health is displayed.
Date/Time	The time and date of the log entry in year, month, day, hour, and minute.
Source	The source service the entry was received from.
Event ID	System defined Event number. User-defined Event ID is defined when used.
Account	Identifies the account, which logged the event.
Message	The detail description of the logged event.
Message Detail	The event detail box provides full descriptive data for a selected event.

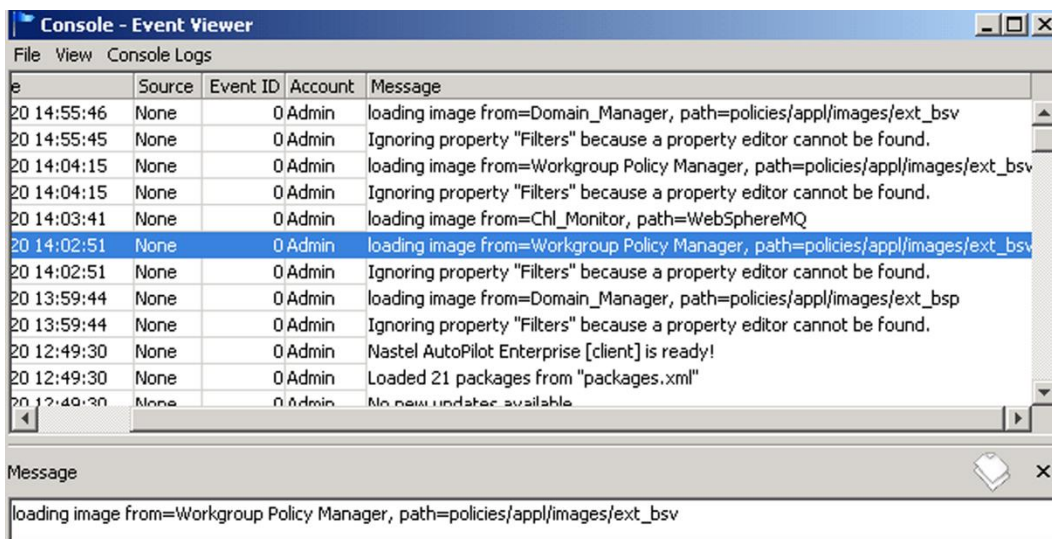


Figure 4-118. Sample Event Viewer and Log

The definable logging characteristics for managers, policies, and experts are common.

Table 4-51. Logging Characteristics	
Property	Description
Audit	Select checkbox to enable the Audit Trace option. The audit trace is used when finite accountability of object, process, and user actions is required. The information collected will be logged in the event file specified in the associated experts.
Log Name	The name of the system-assigned default event log where events are recorded. You should provide a logical name for each policy/manager to prevent the date from being logged in the generic system logs.
Log Service Activity	Click the disable/enable button to disable or enable the log service activity. Disabling the activity will prevent the associated events from being logged in the event logs. If the logging for a manager is disabled, only the Manager's events will be blocked, if the expert logging is enabled, the events generated will be logged in the file specified.
Log size	Log size in bytes. Real log size is the maximum value of server.log.size and logsize.

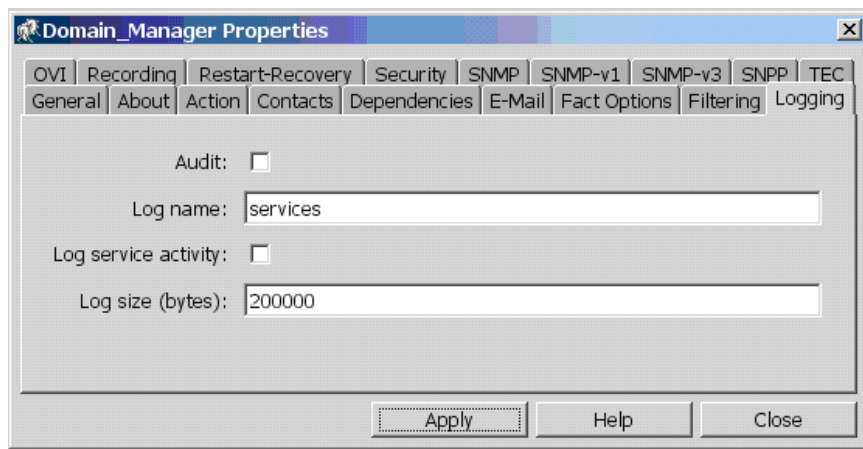



Figure 4-119. Logging Characteristics

4.11.1 Viewing Event Logs

You can view event logs from the event viewer or as text files using most text editors. Each business view logs all events, errors, triggers, and actions it executes into an event log. To open the history of an event for a specific business view:

1. Open your business view.
2. Click  **Event Viewer** to open the viewer at the bottom of the business view screen.
3. Click **File**, and then click **Open** to display the listing of available logs.

4. Click the log that matches the business view selected in step 1 to display the log.

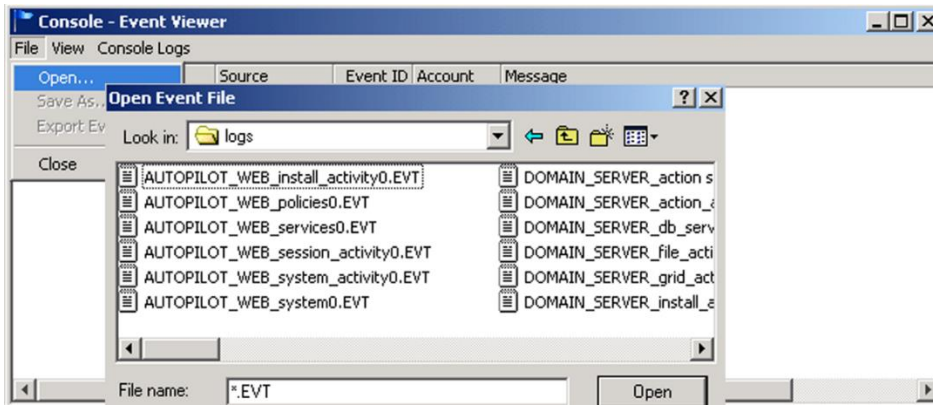



Figure 4-120. Opening Event Log in Event Viewer

The information presented in the events is the same as when viewed in a standalone event view or in the business view event viewer. Event data is formatted as follows:

Table 4-52. Event Logs	
Property	Description
Severity	Depicted with the relative icon that represents the current health
Date/Time	Displays the date and time of the log entry
Source	Identifies the source of the entry as applicable
Event ID	Provide available event numbers
Account	Name of the account which logged the message or event
Message	Contain descriptive information about the event. Contents will vary with the nature of the sensor.
Event Detail	Provide detailed information about the event, source, and status. To open the detail window, double-click the event you want to view. You can copy the detail by clicking the Copy All  icon at the lower right of the screen.

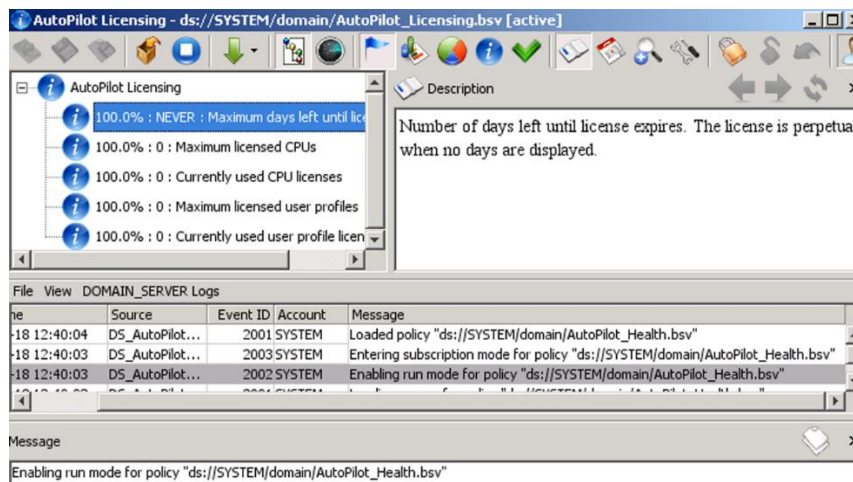


Figure 4-121. Sample View of an Event Log

4.11.2 Event Log Options

Saving and Exporting Log Files

The content of the event logs can be read in any text editor with the limitation that the time/date stamp is specified in milliseconds since it is an OS-specific date. Exported **.TXT** file contains events in comma-delimited format, in which time is translated into absolute date-time. See the sample entry below:

```
2003-05-08 17:21:56, time=1052428916619, type=8, evid=1001, account=Admin, from=Service, event=Starting service XYZ,
```

You may want to save a particular log for later use. You can use the *Save As* or *Export Events* functions to archive the current log.

Save As: With the event log open, select **File > Save As**. When the *Save Event As* screen is displayed, assign a file name to the log.

Export Events: With your event open or closed, select the **File > Export Events**. The events exported will be a snapshot that reflects the (current) status of the event log.

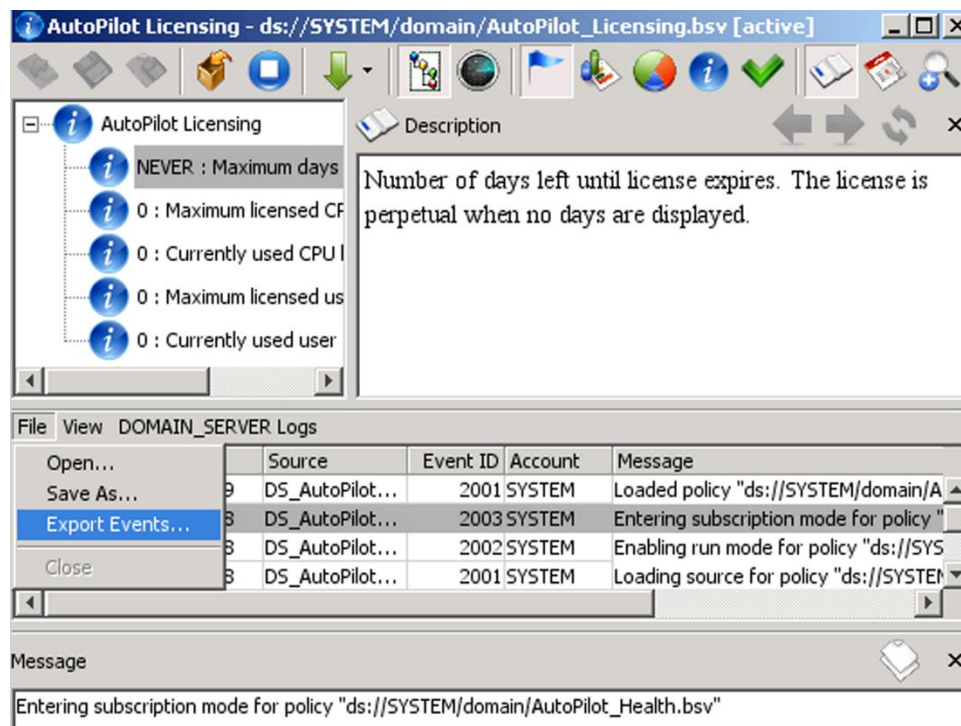


Figure 4-122. Event Log File Options

Viewing and Sorting Event Logs

Event Log viewers have various options that will allow you to customize your logs to best suit your needs. You can use filtering, sorting, and search tools to help identify and define log contents you need.

Event Log Viewing Options: The event log view can be altered to suit your needs by selecting from the following options:

Table 4-53. Event Log View Options	
Property	Description
All Events	Displays all events in the log
Filter Events	Allow you to filter events to display only those events you require.
Newest First	Sort events chronologically from newest to oldest
Oldest First	Sort events chronologically from oldest to newest
Find Event	Use to locate events by severity, ID, and source.
Event Detail	Opens the event detail dialog box at the bottom of the event viewer. The viewer will remain open until manually closed. Provides detailed information about the event, source, and status. To open the detail window, you can also double click the event you want to view.
Refresh	Manually refreshes the event log if active.
Auto-Refresh	Check the <i>Auto-Refresh</i> button to enable. The auto-refresh will remain active until turned off.



Figure 4-123. Event Log Option Menu

Event Filter: Use the filter to customize the log for your needs. You can filter a specific log or all logs using the parameters below:

Table 4-54. Set Event Filter	
Property	Description
Type	Select severity/health levels to define filter.
From	Specify the date and time to search from. Can only be selected when not being initiated from an existing log.
To	Specify the date and time to search to. Can only be selected when not being initiated from an existing log.
Source	Select one or all event logs to filter for results.
Account	Account name. A blank field will reflect all accounts.
Event ID	Specify event IDs to filter. No entry will reflect all events. (Example: 1003)
Event Filter	Name based on event message type to be filtered (example: Starting*). "Starting" will list only those messages that begin with starting. No entry reflects all messages.

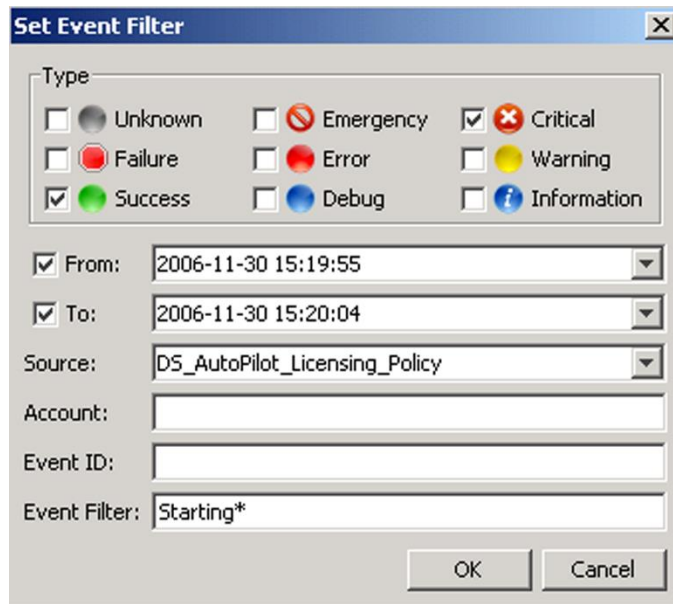


Figure 4-124. Set Event Filter

Find Events: Use this tool to search all event logs or a specific event log using the parameters below:

Table 4-55. Find Event	
Property	Description
Type	Select severity/health levels to define search criteria.
From	Specify the date and time to search from. Can only be selected when not being initiated from an existing log
To	Specify the date and time to search to. Can only be selected when not being initiated from an existing log
Source	Select one or all event logs to filter for results.
Event ID	Specify event IDs to filter.

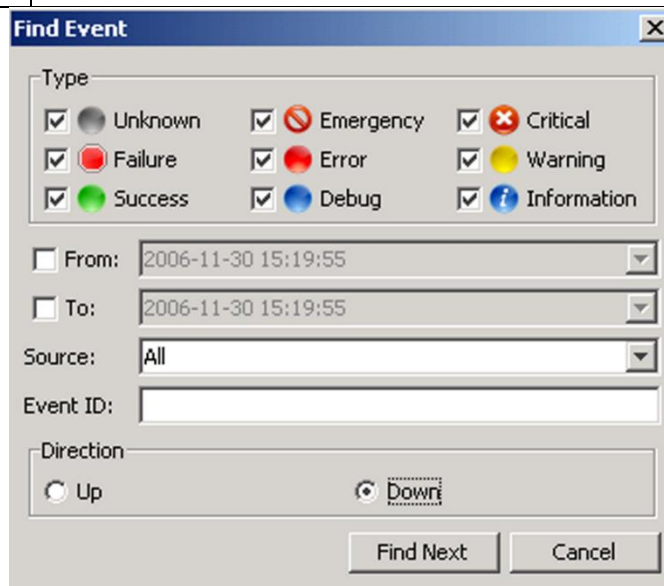



Figure 4-125. Find Event

Log Viewing and Setting Options

 **NOTE** Only one log can be opened in a viewer. If you need to view multiple logs, open additional instances of the event viewer.

The Console Log menu options are only available on the event viewer. This menu is not accessible from the business view event viewer.

Table 4-56. Log Viewing and Setting Options	
Property	Description
Event Log	The <i>Console Logs</i> menu lists the active event logs that are available for viewing. Click the log you want to view to open. (Example: SESSION_ACTIVITY, SYSTEM, UNNAMED).
Clear All Events	Use to clear your log of all events.
Log Settings	Use to set the maximum event log size, denoted in KB. Min= 100KB, Max= 10000KB (10MB)

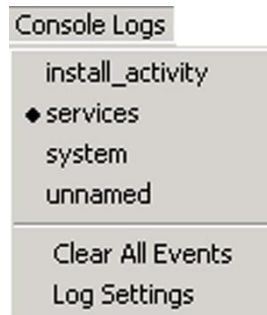


Figure 4-126. Console Logs



Figure 4-127. Log Size Settings

4.12 Performance Monitoring



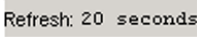





4.12.1 Creating Chart Profiles



New charting profiles can be developed at any time. The performance and operation of Core Services is unaffected by the development of profiles or performance monitoring.

1. Open a performance monitor by clicking the **Performance Monitor**  button.



The performance tool will only allow facts with numeric values to be listed in the table.

2. Select facts, one at a time, to be monitored from the hierarchal menu. Click  **Add** to add the highlighted fact to the table. The monitor will create legend entries.
3. Repeat for each fact you want to monitor. Optionally, click  **Delete** to delete a highlighted fact from the table. The monitor will delete the legend entry.
4. Set the refresh rate and the duration points. The refresh cycle can be set from one to 60 seconds . The duration can be set from 50 to 300 points . Each point represents a refresh cycle that will be displayed on the chart screen. (Example: Refresh Cycle of 10 seconds and Points set at 30 equal five minutes of charting time displayed). The time references at the bottom of the chart reflect the time frame of the current chart screen.
5. Save   the chart profile if you intend to use it again. Give your chart profile a logical name that reflects its nature. If you close the performance monitor without saving the profile, it will not be saved.
6. Start by clicking the **Start Charting**  button.
7. Check the *Available* column in the chart table to ensure the facts are active. Inactive facts, even though they are listed in the table and legend, will not be plotted. A green icon  is displayed when the fact is available for charting. Items that are not available will not be charted.

- If it is necessary to make changes to an active chart, you must pause the chart by clicking the **Stop Charting**  button. The facts available status indicator  will show red when paused. Make the change or addition and then restart when you have completed your changes.

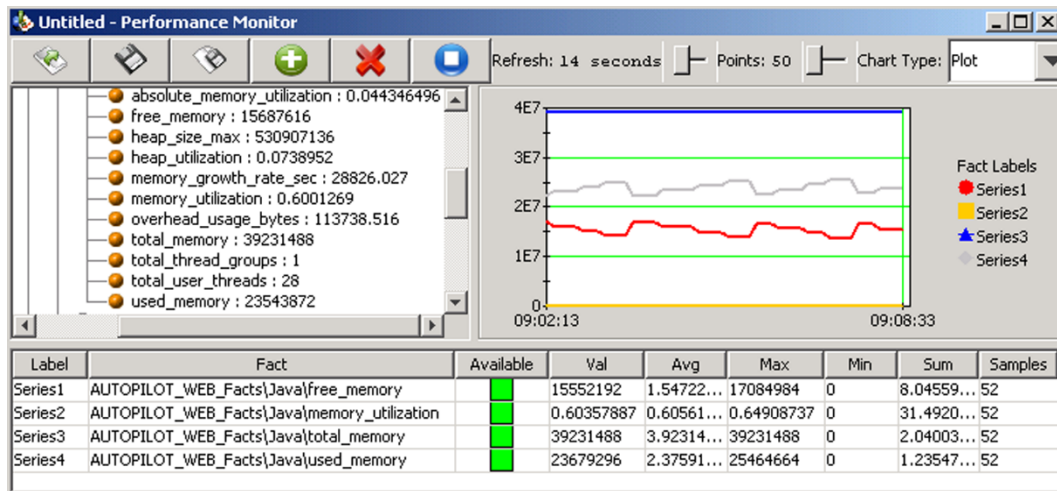









Figure 4-128. Typical Performance Monitor

4.12.1.1 Using Existing Chart Profiles

Use existing chart profiles to repeat monitoring the same performance statistics, or use an existing profile as the template for a new profile.

- From the User Console open the Performance Monitor.
- Click  **Open Chart Profile** to access the menu. Select a chart profile file (.pmv). Click the **Open** button.
- Make any required changes, add/delete facts, and change refresh or duration settings. Click  **Add** to add the highlighted fact to the table. The monitor will create legend entries. To remove an item, click the fact; when it is highlighted, click  **Remove** to delete the item. The monitor will update the legend to remove the deleted fact. Set the refresh rate and the duration points. The refresh cycle can be set from one through 60 seconds . The duration can be set from 50 through 300 points . Each point represents a refresh cycle that will be displayed on the chart screen. (Example: A refresh cycle of 10 seconds and Points set at 30 equals five minutes of charting time displayed). The time references at the bottom of the chart reflect the time frame of the current chart screen.
- If changes are made, you should save the change before starting. Unsaved profile setting will be lost. Alternately, use **Save as** to create a new profile, preserving the

original. Click **Save**  or **Save As**  to save the profile or create a new profile, respectively.

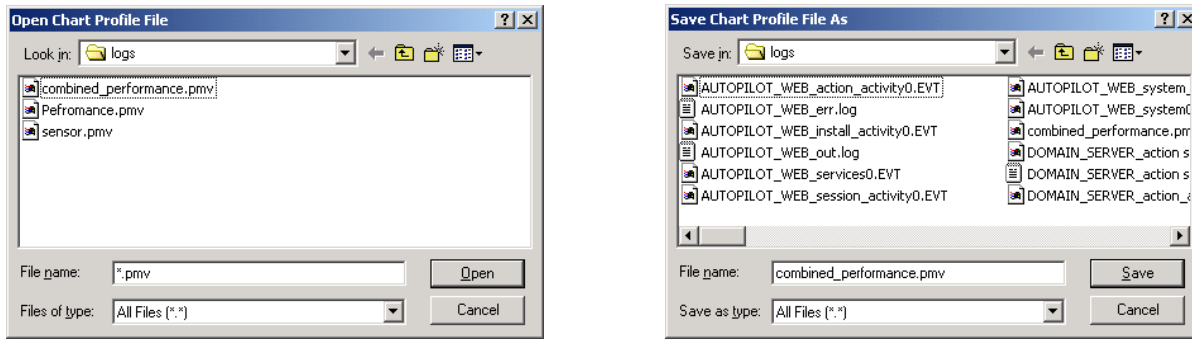




Figure 4-129. Open Chart Profiles and Save Chart Profile As

5. Click the **Start Charting**  button to begin.
6. Check the *Available* column in the chart table to ensure the facts are active. Inactive facts, even though they are listed in the table and legend, will not be plotted. A  green icon is displayed when fact is available for charting. Items that are not available will not be charted.
7. The chart will automatically stop when the user specified chart duration is complete.

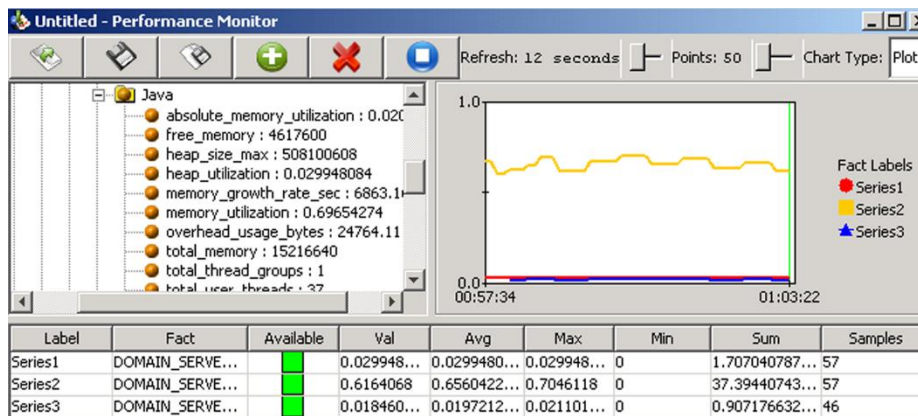


Figure 4-130. Sample Performance Monitor Charting

4.12.2 Monitoring Facts using Performance Monitor

The Performance Monitor starts plotting when activated. The subsequent plots will be posted at the interval you set when the *Refresh* cycle time was set. The total number of plots will be based on the points setting. The plot values are based on the average for the time between refresh cycles. The longer the time between refreshes, the less detailed the chart. Short duration events such as memory or processor usage spikes will be averaged, and each event may not be shown in the graph. When short refresh cycles are used, the detail gives a more accurate depiction of actual events.

The numeric charting parameters are taken from the facts selected for monitoring. The chart will adjust for each fact added to the table. For example, if a plot is set to refresh every four

seconds, it will show a more accurate performance measurement compared to a plot with a refresh rate of every 30 seconds, which only allows for progressive changes to be shown.

Time on left denotes the time the expiring segment or beginning of chart was recorded. Time on right specifies current time (based on internal clock of host machine). The time span between the noted times is the duration set when you set the refresh and the points.

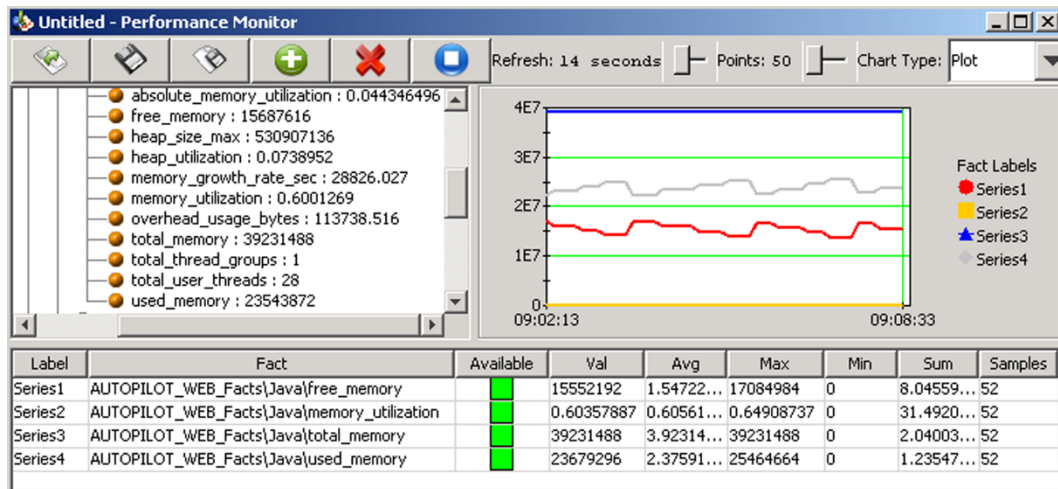


Figure 4-131. Performance Chart (Plot Format)

Results of monitoring are posted in the chart table. They reflect cumulative values since the chart was initiated.

Label	Fact	Available	Val	Avg	Max	Min	Sum	Samples
Series1	DOMAIN_SER...	█	3984944	3510179.2	4099040	0	1.7550896E7	5
Series2	TECHWRITER...	█	1649032	1659010.0	1797448	0	6636040.0	4
Series3	TECHWRITER...	█	4947.2	-3813.666750...	6820.533	-13511.2	-15254.66700...	4
Series4	TECHWRITER...	█	0.66919005	0.6671883625	0.7207304	0	2.66875345	4

Figure 4-132. Chart Table

There are five charting formats available: *Plot*, *Bar*, *Stacking Bar*, *Area*, and *Stacking Area*. Click the format in *Chart Type* that best suits your needs. Charting formats can be changed at any time without interrupting the charting process.

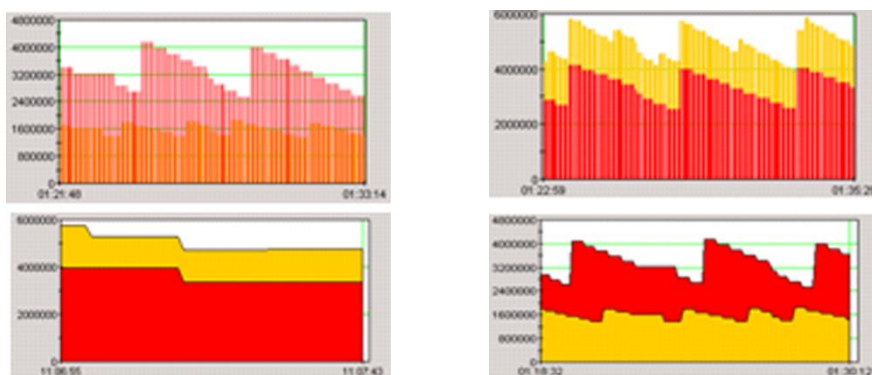


Figure 4-133. Display Chart Formats

4.13 Searching the meshIQ Domain

Search Tool can be used to locate registered management services such as experts, managers, policies, services, and so on, deployed across the meshIQ domain based on user-defined search criteria. Search will also look within managers for content if option is selected.

Table 4-57. Search Tool Properties

Property	Description
Search Parameters	
Name	Enter exact name of Object to be found.
Description	Enter a description of the object to be found.
Role	Associated role of the service. See properties for applicable values.
Property	You can search by using one of the following properties: node, grid_enabled, grid_name, updated, status, auto start, stream facts, stream derived metrics, description, uniqueness, resourceHandle, resourcePath, popup_menu_class, popup_window_class, oid, class, roles, owner, and umask. Wildcard (*) can be used as appropriate.
Context	Select the context from the sub-menu. The available context entries reflect the actual context of the existing objects.
Type	Select the type from the sub-menu. The available type entries reflect actual types of existing objects.
Find managers with such contact	Check the box to perform an internal search of Managers to locate additional objects.
Search Result Table	
Name	Names object found in the search.
Context	Specifies context of object found.
Type	Identifies the type of object found in the search.
Node	Identifies CEP server or domain server where object was located.
Owner	Identifies owner of object found in the search.

Right-click any object listed in the results table to access a context-sensitive menu.

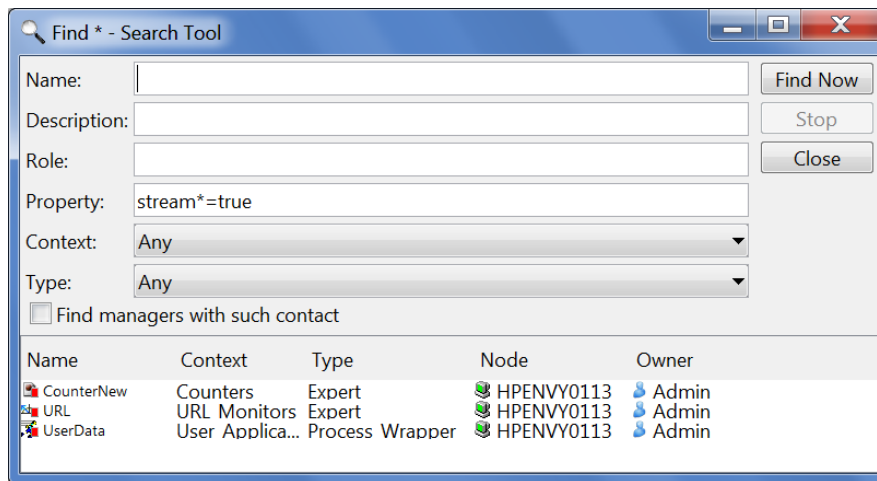


Figure 4-134. Search Tool and Results

4.14 Forwarding Events to OVO from Business Views



An understanding of Business Views and HPOV operations is required to use this section of the manual. Please refer to the Business View sections of this manual and to the pertinent HP OVO documentation for details on these subjects.

Alerts generated by the business views can be forwarded to OVO in several methods. One of these methods is described below.



As shown above, an `opcmsg` command is issued by the business view which is received by the OVO Agent and formatted as per the relevant message template. Message templates typically match the parameters in the issued `opcmsg` and/or the pattern in the message text and format it for display and/or issue actions to notify the relevant personnel and generate trouble tickets.

The typical steps are as follows:

1. Sensors in business views generate alerts based on the monitored facts.
2. The sensor passes certain variables and calls the `opcmsg` command with certain parameters, either directly or through a script. These parameters include severity, object type, application, and message text. The object type and application are based on the facts being monitored. The name of the sensor is usually passed as the message text.
3. The OVO agent running on the CEP server intercepts this call and matches it with the message templates deployed on the node.

4. If the message pattern and parameters match with any of the conditions in the template, then the message is formatted for display on the browser (active or history); used to auto-acknowledge a previous event; highlight the service tree; notify the responsible parties and/or open a trouble ticket.

Use the message template defined by the file `bsv_interface.dat`. This is a message template that can be used by alerts generated from business views. Please contact your OVO administrator for procedures and guidance.

4.14.1 Issue `opcmsg` with a Message Text

You can issue an `opcmsg` with a message text in one of the following formats:

For Agent related alerts:

```
AP_BSV::<*.gm>::<*.node>::<*.objtype>::<*.property>::<*.value>::REASON
```

```
Ex: AP_BSV::GM::NODE::wmqagt::state::disconnected::EXRC_AGENT_DISCONNECTED
```

The possible reason codes are:

```
EXRC_AGENT_DISCONNECTED
```

```
EXRC_AGENT_CONNECTED
```

For Queue Manager related alerts:

```
AP_BSV::<*.gm>::<*.node>::<*.objtype>::<*.property>::<*.value>::REASON
```

```
Ex: AP_BSV::GM::NODE::qmgr::state::Inactive::EXRC_QMGR_STOPPED
```

The possible reason codes are:

```
EXRC_QMGR_ACTIVE
```

```
EXRC_QMGR_STOPPED
```

For Object related alerts:

```
AP_BSV::<*.gm>::<*.node>::<*.qmgr>::<*.objname>::<*.objtype>::<*.property>::<*.value>::REASON
```

```
Ex: AP_BSV::GM::NODE::QMGR::APPL.INPUT.QUEUE::queue::Curdepth::15::QDEPTH_THRESHOLD_REACHED
```

The possible reason codes are:

```
EXRC_CMDSRVR_STARTED
```

```
EXRC_CMDSRVR_STOPPED
```

```
MQRC_CHANNEL_STARTED
```

```
MQRC_CHANNEL_STOPPED_ERROR
```

```
MQRC_CHANNEL_STOPPED_RETRY
```

MQRC_CHANNEL_STOPPED_OK
 MQRC_CHANNEL_STOPPED_DISABLED
 MQRC_Q_DEPTH_FULL
 MQRC_Q_DEPTH_HIGH
 MQRC_Q_DEPTH_LOW
 QDEPTH_THRESHOLD_REACHED
 QDEPTH_THRESHOLD_NORMAL

4.14.2 Examples of Sending Alerts

The following are examples of sending an alert when a queue reaches a depth threshold. The alert screen below is from a dynamic sensor that looks at the curdepth fact and generates an alert if the depth is greater than n.

Example 1: The alert will be sent to OVO and can be manually acknowledged by the operator.

The screenshot shows the 'Properties' dialog box with the 'Alert' tab selected. The 'Alert on' section has radio buttons for 'Unknown', 'Failure', and 'Success', and checkboxes for 'Emergency', 'Error', 'Debug', 'Critical', 'Warning', and 'Information'. The 'Alert options' section includes a checked 'Suppress false alarms' checkbox and a 'Use Defaults' button. Below are fields for 'From user', 'Mail server (SMTP)', 'E-mail to', 'E-mail subject', 'E-mail MIME type', 'User action', 'Event ID', 'Event mask', 'Scheme', 'Repeat limit', 'Trigger delay (ms)', and 'Re-arm delay (ms)'. The 'User action' field contains a complex command: `plication=apwmq "msg_text=AP_BSV::%f0(1)%::%f0(2)%::%f0(3)%::%f0(4)%::queue::CurDep`.

Figure 4-135. Example of Queue Depth Manual Acknowledgement Alert Screen

The possible severities are Critical and Warning. The following alert is issued:

```
opcmsg severity=%sevstr% object=queue application=apwmq
"msg_text=AP_BSV::%f0(1)%::%f0(2)%::%f0(3)%::%f0(4)%::queue::CurDepth:
:%value%::QDEPTH_THRESHOLD_REACHED"
```

Example 2: Alerts will be auto-acknowledged when the business view turns to success.

The business view in the above example can be modified to issue the following alert:

```
object_alert_ovo.bat %sevstr% %f0(1)% %f0(2)% %f0(3)% %f0(4)% queue Curdepth
%value% QDEPTH_THRESHOLD_REACHED
```

The script `object_alert_ovo.bat` converts the incoming parameters to severity, application, object, and message text.

The script follows:

```
REM *
@echo
REM *
set SEVSTR=%1
set GM=%2
set NODE=%3
set QMGR=%4
set OBJECT=%5
set OBJ_TYPE=%6
set PROPERTY=%7
set VALUE=%8
set REASON=%9

REM *** If business view issues SUCCESS Severity and REASON must be changed
accordingly*
IF %SEVSTR%==SUCCESS goto CHG_REASON
goto ISSUE_ALERT

:CHG_REASON
set SEVSTR=NORMAL
IF %REASON%==EXRC_CMDSRVR_STOPPED set REASON=EXRC_CMDSRVR_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_ERROR set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_RETRY set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_OK set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_DISABLED set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_Q_DEPTH_FULL set REASON=MQRC_Q_DEPTH_LOW
IF %REASON%==MQRC_Q_DEPTH_HIGH set REASON=MQRC_Q_DEPTH_LOW
IF %REASON%==QDEPTH_THRESHOLD_REACHED set REASON=QDEPTH_THRESHOLD_NORMAL

:ISSUE_ALERT
```

```
echo opcmmsg severity=%SEVSTR% object=%OBJ_TYPE% application=apwmq
```

```
msg_text="AP_BSV::%GM%::%NODE%::%QMGR%::%OBJECT%::%OBJ_TYPE%::%PROPERTY%::%VALUE%::%REASON%"
```




The method described above is only one of several methods. You should modify it per your own requirements.

4.15 Streaming Data

To stream to the tracking application, you can use <https://xray.meshiq.com/xray> or your local tracking installation.

4.15.1 Stream Data

In order to stream data, modify the `global.properties` file as follows:

- Specify the access token. You can copy the token from the in one of two ways:
 -  > Admin Settings > Tokens > Streaming
 -  > Admin Settings > Tokens > Repositories. Click the Setup icon  in the Streaming Tokens column.
- (Optional.) Specify the data center and the geo location.
- Specify the URL of the streaming service. For your on-premises installation, replace `https://data.jkoolcloud.com` and enter the location of your local cluster (for example, `http://localhost:6580`).

```
; Nastel XRay specific options
; set following to true to enable shutdown statistics dump
property tnt4j.dump.on.vm.shutdown=false
property tnt4j.dump.on.exception=true
property tnt4j.dump.provider.default=true
property tnt4j.config={autopilot.home}tnt4j.properties
property tnt4j.token.repository={autopilot.home}tnt4j-tokens.properties
property tnt4j.source.DATACENTER=your_datacenter ◀ 2
property tnt4j.source.GEOADDR=your_geo ◀ 2
; uncomment if streaming events or metrics to Nastel XRay
;property tnt4j.source.hosturl=https://data.jkoolcloud.com
;property tnt4j.source.access.token=your_accesstoken
; uncomment if streaming support data to Nastel XRay
;property tnt4j.support.hosturl=https://data.jkoolcloud.com ◀ 3
;property tnt4j.support.access.token=your_support_accesstoken ◀ 1
; control type of NRD files created automatically
property server.dump.format.json.suppress=true
property server.dump.format.nrd.suppress=false
property tnt4j.default.event.factory=com.jkoolcloud.tnt4j.sink.impl.slf4j.SLF4JEventSinkFactory
include ../wgs11.properties
```

These settings along with those specified on the expert properties and `tnt4j.properties` are used to control the streaming options.

4.15.2 Stream Facts

In order to stream Core Services facts (metrics), you configure the destination and which facts to stream. Keep in mind that in the tracking application, Core Services facts become *snapshots*.

1. Right-click on any core service such as an expert or a policy manager and select **Properties** to open the Properties dialog box. You can also set these properties when deploying a new service.

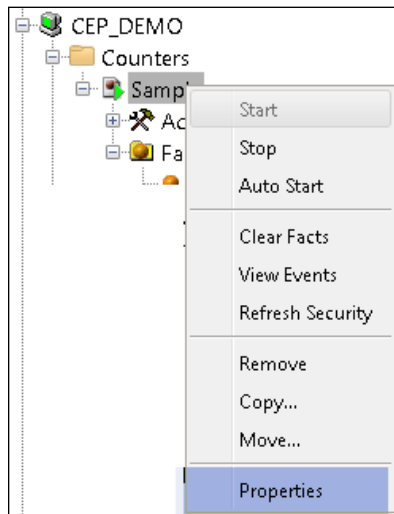


Figure 4-135A. Service Properties

2. On the **Streaming Options** tab, configure the options required as described below. The **Streaming Options** tab controls which facts get streamed and the location they stream to. Many of the fields are optional and have default values provided in the `tnt4j.properties` file that is part of the data services installation. However, they can be overridden.
 - a. For streaming the facts as they update, select **Stream Facts**.
 - b. For streaming periodic derived metrics, select **Derived Metrics** and set a frequency by entering a time interval in milliseconds in the **Interval of derived metrics** field.

NOTE: It is recommended to set the exclude/include filter to control which facts are streamed.

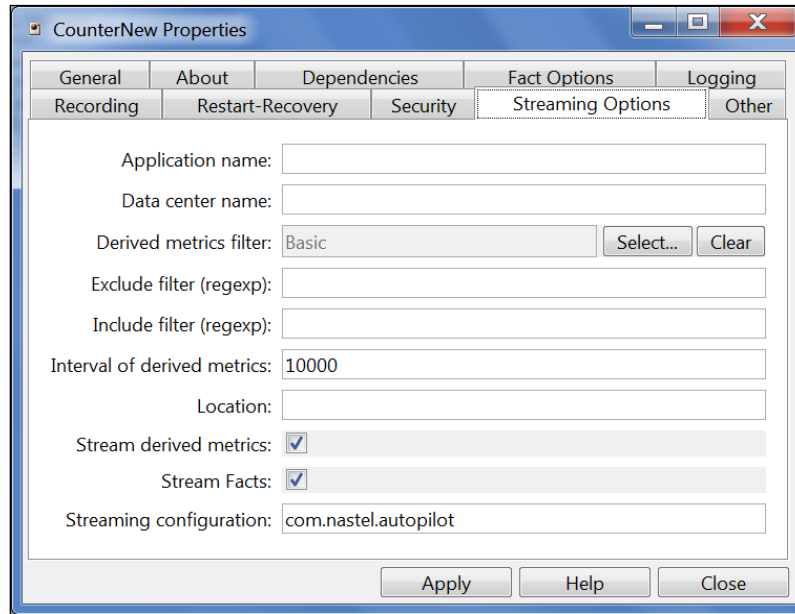


Figure 4-136. Streaming Options

Table 4-58. Streaming Options Properties	
Property	Description
Application name	Sets application name if different from the default set in the <code>tnt4j.properties</code> file.
Data center name	Sets data center name if different from the default set in the <code>tnt4j.properties</code> file.
Derived metrics filter	Click Select to select an existing filter or create a new one. (See Figure 4-137 .)
Exclude filter (regex)	Ignore facts that match specified regular expression; that is, do not stream facts that match the regex.
Include filter (regex)	Only stream the facts that match specified regular expression.
Interval of derived metrics	Time interval, in milliseconds, to send fact derived metrics.
Location	Sets server location if different from the default set in the <code>tnt4j.properties</code> file.
Stream derived metrics	Enable/disable derived metrics streaming. System facts are generated: <code>total_derived_fact_processed</code> and <code>total_derived_fact_sent</code> .
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework). System facts are generated: <code>total_fact_processed</code> and <code>total_fact_sent</code> .
Streaming configuration	Indicates where the data streams. This value must match a stanza in the <code>tnt4j.properties</code> file. The default is com.nastel.autopilot .

Create a New Filter

Filters are used to specify which derived metrics you want to stream.

1. After clicking the **Select** button from the **Streaming Options** tab (Figure 4-136), the *Filters* dialog box is displayed. Click the plus sign icon to display the *New Filter* dialog box (Figure 4-138).

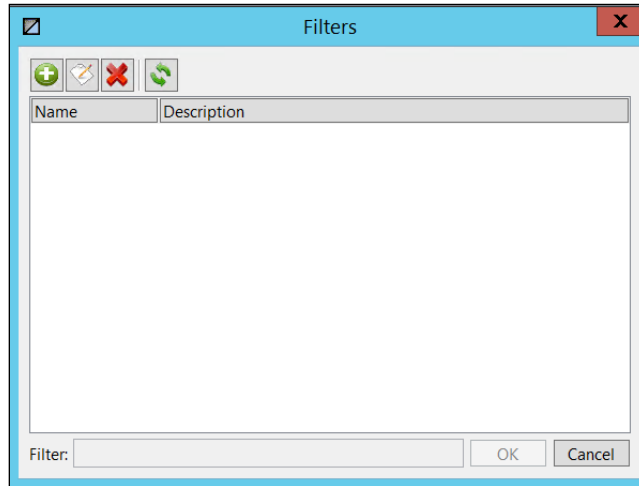


Figure 4-137. Create a Filter

2. From the *New Filter* dialog box:
 - **Filter name** – Enter a name for the filter.
 - **Derived metrics** – Select **General**, **History**, **Statistics**, or **All** to get a list of related metrics.
 - **Description** – Enter a description for the filter.
 - **Available metrics** – Select a metric and use the **Add** button to move it to the **Selected metrics** column on the right. You can use **Add all** to move all the metrics. Similarly, you can use **Remove** and **Remove All** buttons to remove metrics. (Refer to [Appendix G](#) for a list of derived metrics.)

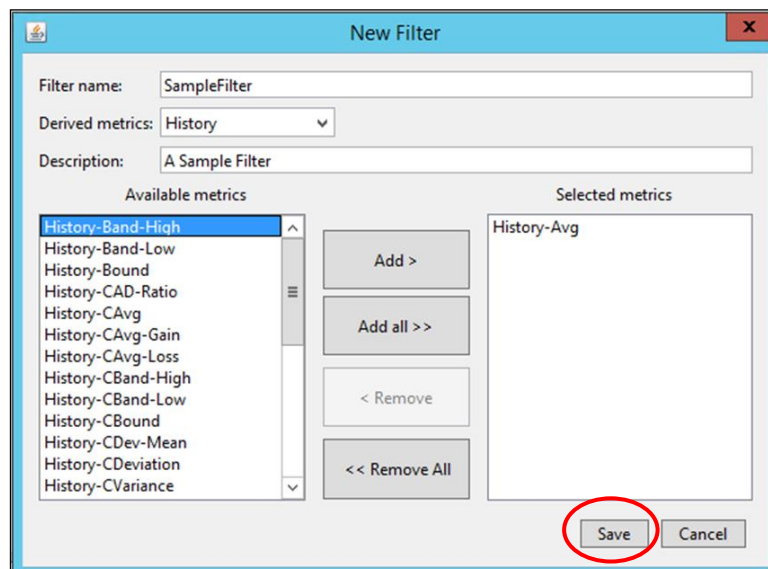


Figure 4-138. New Filter

3. Click **Save** to save/create your filter. It is now listed on the *Filters* dialog box ([Figure 4-139](#)) and can be selected for future use.

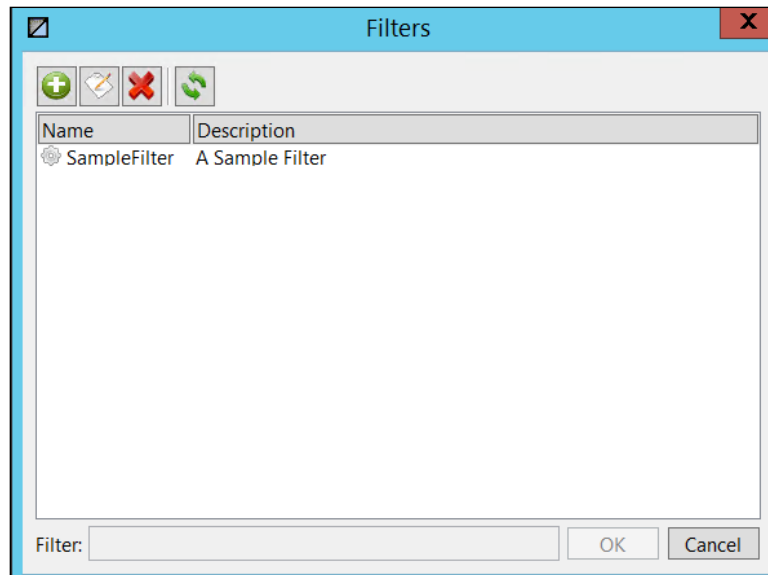


Figure 4-139. Filters Dialog Box

4.15.3 Logging Policies

Keep in mind that in the tracking application, Core Services policies results become *events*. Before streaming events, the steps in 4.15.1 must be completed. Then, the log4j2.xml file must be configured to forward AutoPilot policy and other events.

```
<!-- uncomment the following to stream policies events to Nastel XRay -->
<!--Logger name="policies" level="info" additivity="false">
|   <AppenderRef ref="Tnt4j"/>
</Logger-->

<!-- eventstream events are streamed to Nastel XRay -->
<Logger name="eventstream" level="info" additivity="false">
|   <AppenderRef ref="Tnt4j"/>
</Logger>
```

To stream all policy events, uncomment the 3 lines shown by removing “!--” from the beginning and “--” from the end as shown below

```
<!-- uncomment the following to stream policies events to Nastel XRay -->
<Logger name="policies" level="info" additivity="false">
|   <AppenderRef ref="Tnt4j"/>
</Logger>

<!-- eventstream events are streamed to Nastel XRay -->
<Logger name="eventstream" level="info" additivity="false">
|   <AppenderRef ref="Tnt4j"/>
</Logger>
```

You can repeat this process for any service logs you want to stream, as shown in the example for eventstream.

The [Logging tab](#) can be configured to specify the log file. The **Logging** tab can be found on the **Property** dialog box for the policy manager or for specific policies by accessing the **Properties** tab for it. The latter is the option which will stream by default (policies).

1. Right-click the policy manager or specific policy and select **Properties** from the drop-down menu.

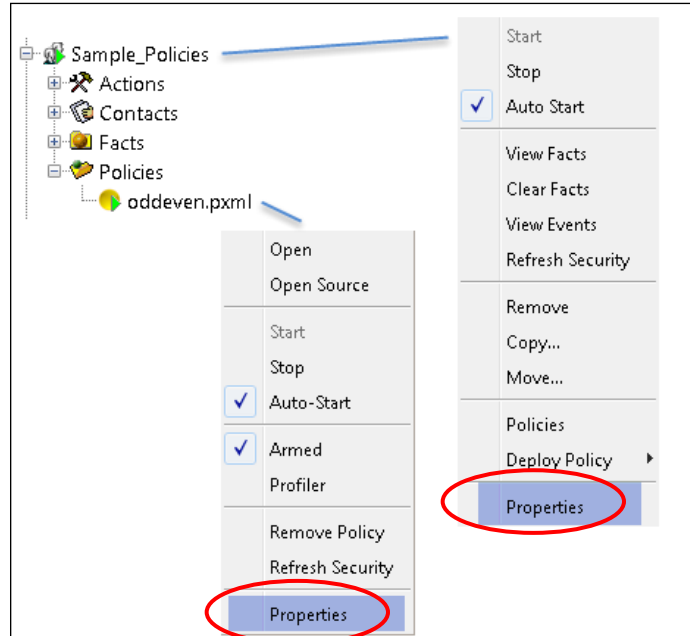


Figure 4-140. Policy Manager and Policy Menus

2. The **Logging** tab is identical regardless of which method is used. The values specified here for **Log name** must match the value specified in **log4j2.xml** in order to stream.

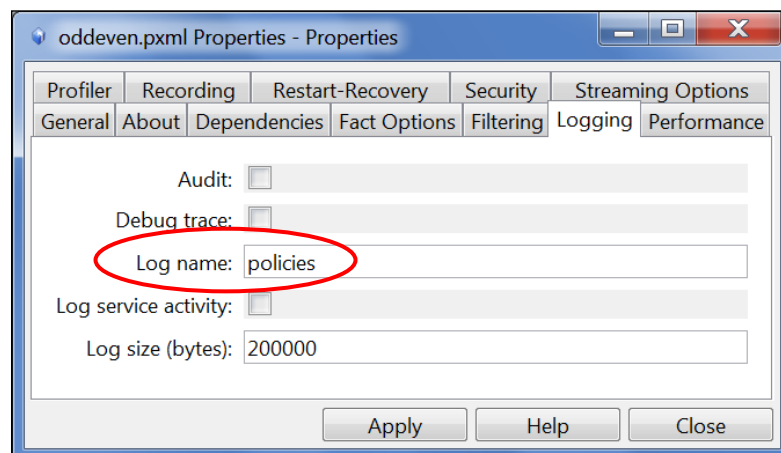


Figure 4-141. Logging Tab

Core Services Events are represented in the tracking application as Activities, Events, Snapshots, and properties.

The activity represents the “event” generated by Core Services and ties the event activity together. The events represent the different types of actions that Core Services can take as a result of an event, including sending email, running methods, and executing user actions. The snapshot contains the parameters associated with the state of the Core Services event.

Chapter 5: Customizing Core Services

This chapter describes customization options of various components. Customization should be performed by a meshIQ Platform administrator, who must have admin privileges to the file system where Core Services are installed.

This section applies to components such as the CEP server, domain server, and User Console.

5.1 Overview

Customization can be divided into the following categories:

- **Operational Properties** control the operation run-time behavior of each component. Operational properties are maintained in `node.properties` and `.LAX` files.
- **Performance Properties:** Alters performance behavior of components. Each component (CEP server, domain server and User Console) is customized via **`node.properties`** files and corresponding `ATP*.LAX` file. There is a separate **`node.properties`** file for details. (Refer to table below.)
- **Registry Configuration:** all deployed service definitions are maintained in registry files. Registry files are `.XML` files, which are dedicated for each domain server, CEP server and web server. Registry files can be moved from one CEP server to another for migration purposes. They can also be merged by placing a registry file into a corresponding **`import`** directory. Once merged by a CEP server, the registry file is renamed into an **`import\<file>.xml.merged`**. Registry files with `.DAT` extension are binary files and do not follow the same rules as the XML files.



NOTE

You can use the “include” command to include property files within other property files. For example, `include ../plugin.properties`.

Table 5-1. Server Property Files (apwmmq.properties)

Components	Core Services Properties	JVM Properties Files
Domain Server	<code>naming\node.properties</code>	<code>ATPNAMES.LAX</code>
CEP Server	<code>localhost\node.properties</code>	<code>ATPNODE.LAX</code>
User Console	<code>mconsole\node.properties</code>	<code>ATPCONS.LAX</code>
Web	<code>apache-tomcat\webapps\autopilot\node.properties</code>	N/A

Table 5-2. Server Registry Files		
Registry	Locations	Description
registry.xml	Domain Server: \naming CEP Server: \localhost User Console: \mconsole Web: \apache-tomcat\ webapps\autopilot	Contains all deployed service definitions for the corresponding server or console. It includes experts, managers, and policies.
naming.xml	Domain Server: \naming	Contains domain directory registration entries. All registered services in the domain maintained in this file.
security.dat	Domain Server: \naming	Non-XML binary file that contains domain account definitions. File must be secure. Loss or corruption of file will result in loss of all account information.
actions.xml	Domain Server: \naming	Definitions of all globally defined user actions.
fileacl.xml	Domain Server: \naming	Access control list for all user-defined policies (.bsv, .pxml and .bsp files).
filelock.xml	Domain Server: \naming	Policy lock file maintains information about users who hold policy locks.
apwmq.properties	Domain Server: [autopilot_home]	Used to define abbreviations that can be used in fact names of a business view. Example: property GROUP = MESHQPRD property WSMON = WS_Monitor property NODMON = Node_Monitor property QMMON = QM_Monitor (See figure 4-141-b below)

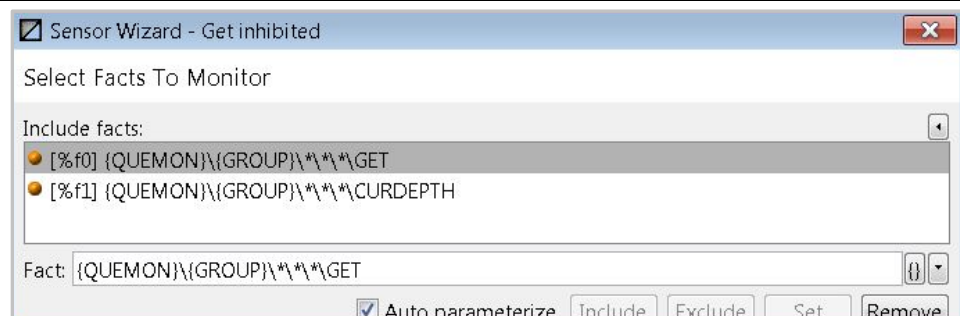


Figure 5-1. Sensor Wizard – Use of Aliases

Registry XML files can be edited by hand with any text or XML editor, for example, . However, this should be done with extreme caution. This could be useful when making global changes that otherwise might require too many console procedures.

Do not edit or remove the `security.dat` registry file. Make sure all registry files are backed up periodically and can be restored in case of a problem, loss, or corruption.

5.2 Java Runtime—LAX File Customization

LAX files are associated with every meshIQ Platform Core Services executable. A process name has a corresponding file with the `.LAX` extension (example: `ATPNODE.lax`). `.LAX` files define Java run-time environment such as JRE, path, `CLASSPATH`, memory limitations and other Java start-up parameters. Most parameters do not require changes, with the following exceptions:

- **`lax.nl.java.option.java.heap.size.initial`:** Initial heap size (bytes) of JVM where the component is executing. The default setting is system dependent. The minimum required is 6 GB; the recommended setting is 16 GB. If CEP server(s) are monitoring large environments (10000 or more facts), default number can be increased.
- **`lax.nl.java.option.java.heap.size.max`:** Maximum heap size (bytes) of JVM where the component is executing. The default heap size is system dependent. If CEP server(s) are monitoring large environments, the default number can be increased.

If properties above are not defined, include them by adding lines to `.LAX` files as follows:

```
// initial heap in bytes
lax.nl.java.option.java.heap.size.initial=64000000
// maximum heap in bytes
lax.nl.java.option.java.heap.size.max=256000000
```



The `lax.nl.current.vm` file points to the location of the Java file.

NOTE

5.2.1 Using Server JVM

For optimal performance, we highly recommend replacing the property **`lax.nl.java.option.additional=-server`** with the following on Windows, Solaris and Linux systems, and enabling it. Replace this property in `ATPNODE.lax` and `ATPNAMES.lax`:

```
lax.nl.java.option.additional=-server -Xnoclassgc -XX:SurvivorRatio=16  
-XX:PermSize=128m -Xmn256m
```

This enables the use of server JVM, which allows meshIQ Platform Core Services to take advantage of multiple CPUs and enhanced memory and CPU performance.

5.3 Server Runtime – PROPERTY FILES

meshIQ Platform Core Services maintains five property files as follows:

- `[AUTOPILOT_HOME]/global.properties` – properties shared by all run-time components such as domain server, CEP servers, console, and web server (if applicable).
- `[AUTOPILOT_HOME]/domain.properties` – global properties shared by all CEP servers within a domain. File is loaded by domain server only and cannot be refreshed while domain server is running. Domain server must restart to apply changes. It is recommended to use “shared.” prefix for all properties to distinguish global properties from local ones. For example: `shared.prop1=test`
`shared.prop2=test2`
- `[AUTOPILOT_HOME]/naming/node.properties` – properties shared by domain server and all deployed services within the domain server.
- `[AUTOPILOT_HOME]/localhost/node.properties` – properties shared by CEP server and all deployed services within the CEP server.
- `[AUTOPILOT_HOME]/mconsole/node.properties` – properties shared by the Console.

These supported properties can be defined and overwritten by administrators by copying into `[AUTOPILOT_HOME]/global.properties` or individual `node.properties` to take effect.

Each `node.properties` file is identical in structure but may define different properties for various components. This file defines runtime configuration that defines server operational parameters.

Properties are defined as: *property property_name= value* format. User-defined properties can be defined as well.

For example:

```
property server.facts.capacity = 100000
property server.net.sessions.poolsize = 3
property server.net.agents.poolsize = 1000
property server.log.size = 2000000
```

Table 5-3. General Server Properties

Property	Description	Values
<code>autopilot.home</code>	Install directory set automatically during installation. DO NOT change or define this variable.	String. <code>[install_dir]</code>
<code>server.type</code>	Type of CEP server. DO NOT change this property.	String. <code>[type]</code> Server, Domain, Client
<code>installation.update.groups</code>	List of groups that installation is subscribing to for updates.	String. Console, Mandatory, Service_Updates

Table 5-4. Console Properties

Property	Description	Values
console.external.browser	Launch external web browser within business views.	Boolean. Default is true.
console.pwd.notice	Early expiration of password notification.	Boolean. Default is true.
console.pwd.notice.days	Number of days until password expiration.	Integer. Default is 15.
gui.lookandfeel	Look and Feel of the GUI. Default – default Java look and feel Native – native OS look and feel SkinLF – skinnable look and feel based on supplied theme packs.	Boolean. Default is Native.

To enable one of the theme packs listed below, set gui.lookandfeel to SkinLF and uncomment one line.

```
gui.skinlf.themepack=../themes/aquathemepack.zip
gui.skinlf.themepack=../themes/macosthemepack.zip
gui.skinlf.themepack=../themes/beosthemepack.zip
gui.skinlf.themepack=../themes/bbjthemepack.zip
gui.skinlf.themepack=../themes/whistlerthemepack.zip
gui.skinlf.themepack=../themes/modernthemepack.zip
gui.skinlf.themepack=../themes/themepack.zip
gui.skinlf.themepack=../themes/xplunathemepack.zip
gui.skinlf.themepack=../themes/toxicthemepack.zip
```

Table 5-5. Policy Performance Properties

Property	Description	Values
server.policy.buffer.size	Facts are buffered and flushed when limit is reached.	Integer. Default is 500.
server.policy.buffer.grace period	Facts are buffered before flushing after defined grace period regardless of buffer size.	Integer. Default is 1000.
server.sensor.delivery.flowpct	Disables/enables flow control.	Integer. Default is 0.
server.sensor.delivery.batch	Processes sensor events.	Integer. Default is 2000.
server.sensor.db.queue.limit	Maximum queue size of all entries.	Integer. Default is 1000.

Table 5-5. Policy Performance Properties

Property	Description	Values
server.sensor.db.queue.flowpc	The percentage at which logging is re-enabled after reaching 100% utilization. In order to make sure that capacity is not immediately exceeded again when it is turned back on, logging is only re-enabled when the utilization drops down to this threshold. Default: 60%.	Integer. Default is 60.
server.sensor.profiler	Enable sensor profiler globally for computation of sensor profiling metrics.	Boolean. Default is false.
servlet.cache.timeout	Cache timeout in seconds.	Integer. Default is 5 secs.
server.topic.delivery.limit=server.facts.capacity	Maximum undelivered published queue level. Messages are dropped if level is exceeded.	String: server.facts.capacity/10
server.pubsub.filter.asis	If set to true, increases filter specificity which eliminates circular filter references among CEP servers.	Boolean. Default is false.
server.pubsub.remote.lookups	Improves policy performance for policies subscribing to remote facts. If set to false, disables remote fact lookups when topic has not been initialized yet.	Boolean. Default is false.

Table 5-6. Rule Engine Performance Properties

Property	Description	Values
server.sensor.table.size	Maximum number of expected child sensors for each parent sensor.	Integer. Default is 1011.
server.sensor.batch	Maximum sensor in the pipeline before forcing rule execution.	Integer. Default is 20.
server.sensor.idle	Maximum idle time in ms. before forcing rule execution.	Integer. Default is 3000.
server.sensor.rule.thread.debug	Enable rule engine debug mode.	Boolean. Default is false.
server.sensor.process.nulls	Enables/disables node.properties to use null processing within rules for processing fact disconnection.	Boolean. Default is true.
server.sensor.comp.mode	Disables/enables null processing within rules.	Boolean. Default is false.

Table 5-7. Communication Properties

Property	Description	Values
server.agent.timeout	Processing time in ms. allowed for requests executed remotely.	Integer. Default is 35000ms.
server.socket.pipe.trace	Enable trace for all socket communication. (Enabled only for debugging and tracing.)	Boolean. Default is false.
server.classloader.debug	Enable all network class loading trace.	Boolean. Default is false.
server.socket.option.TCP_NODELAY	Enables/disables TCP algorithm; true turns it off, false turns it on	Boolean. Default is true.
server.socket.option.SO_KEEPALIVE	Enable TCP keep alive for server running behind a firewall.	Boolean. Default is false.
server.socket.option.SO_RCVBUF	Override TCP receive buffer size in bytes	Integer. Default is 65536 bytes.
server.socket.option.SO_SNDBUFF	Override TCP send buffer size in bytes.	Integer. Default is 65536 bytes.
server.net.heartbeat	Enable heartbeat exchange. Heartbeat exchanges prevent firewalls from dropping idle connections	Boolean. Default is true.
server.net.heartbeat.interval	Heartbeat interval in ms.	Integer. Default is 60000.
server.net.heartbeat.trace	Enable heartbeat trace	Boolean. Default is true.
server.io.input.buffer.size	Control input buffer sizes.	Integer. Default is 8192 bytes.
server.io.output.buffer.size	Control output buffer sizes.	Integer. Default is 8192 bytes.
server.net.sessions.poolsize	Maximum number of parallel sessions between each CEP server. More sessions will improve performance but will require more memory.	Integer. Default is 3. 2 are minimum.
server.net.agents.poolsize	Number of parallel requests that can be issued simultaneously.	Integer. Default is 1000.
com.nastel.nfc.net.trace	Enable all low-level communication trace.	Boolean. Default is true.
server.net.connection	Override with the specified host name, IP for communication. Should always have the value of the public interface.	String. hostname,ipaddress
server.pipe.processor.limit	Maximum number of queued objects for each socket	Integer. Default is 20000.

Table 5-7. Communication Properties

Property	Description	Values
	connection. Connection is disabled and objects will be dropped when the limit is reached.	
server.pipe.delivery.flowpct	Enables/disables flow control.	Integer. Default is 60.
server.process.timeout	Business Views will unblock and continue after the specified timeout value.	Integer. Default is 1000.

Table 5-8. Service Properties

Property	Description	Values
server.services.facts.expire	Default fact expiration for all services. Should only be used to enable server-wide expiration for all deployed services.	Integer. Default is 0.

Table 5-9. Security Properties

Property	Description	Values
server.security.algorithm	Security algorithm. DO NOT change after user accounts are created.	DSA
server.security.provider	Security profile provider. DO NOT change after user accounts are created.	Sun
server.aip.port	Enables AIP communication support. The property can be added to any CEP server's node.properties file.	Integer. Default is port_number.
server.aip.trace	Enables AIP trace.	Boolean. Default is True.
server.security.storage.provider	Security profile storage provider. DO NOT change.	String. com.nastel.nfc.security.SecurityBoard
server.unmask	Default user permission mask applied to every new user account.	String. Base/Control Base/Read Base/Change Base/Execute, Base/Read
server.security.token.expiry	Security session token expiration in ms.	Integer. Default is 10000ms

Table 5-9. Security Properties		
Property	Description	Values
server.security.password.length	Minimum password length.	Integer. Default is 5.
server.jaas.realms	Space-oriented list of domain names that CEP servers and console clients may authenticate with.	String. [Domain 1] [Domain 2]
java.security.krb5.conf	Defines properties for default realm.	String. {autopilot.home}krb5.conf

Table 5-10. Domain Server Properties		
Property	Description	Values
server.domain	Logical name for the domain serviced by the domain server.	Boolean. Default is DOMAIN.
server.naming.url.port	TCP port for the Domain Server directory service.	Integer. Default is 2325.
domain.server.url	URL of the Domain Server where port should equal value of server.naming.url.port	String. Default is: bbns://localhost:port
domain.server.name	Name of domain server as registered in the directory service.	String. Default is DOMAIN_SERVER.
domain.server.failover.url	Failover domain server location. Edit and uncomment the following line to enable domain server failover where port should equal the value of the alternate server.naming.url.port .	String. Default is: bbns://failover-server:port
installation.update.folder	URL location of all meshIQ Platform Core Services software for distribution to all meshIQ installation. ds://equals the root directory in business.view.dir or file.system.root properties.	String: Default is ds://software

Table 5-11. Server Properties		
Property	Description	Values
server.work.dir	Working directory of running server. Read Only.	String. [aphome]/naming
server.user.url.port	TCP port for server-to-server communication.	Integer. Default is 3005.

Table 5-11. Server Properties

Property	Description	Values
server.log.dir	Logs directory where all logs are being written.	String. [aphome]/logs
server.import.dir	Server registry import directory.	String. [workdir]/import
server.snmp.mibs	Location of all SNMP mibs.	String. [aphome]/mibs
server.template.dir	Location of service templates.	String. [aphome]/templates
server.debug	Server wide debug mode for extra information.	Boolean. Default is false.
server.sampling.rate	Sampling rate for server wide statistics for [SERVER]_Facts expert.	Integer. Default is 30000.
server.auto.gc.memory.usage	Force garbage collection when memory utilization hits specified number.	Integer. Default is 0.82.
server.auto.gc.timeout	Minimum time before auto gc cycles.	Integer. Default 600000.
server.facts.capacity	Maximum number of facts/properties that can be handled by CEP server. CEP server may handle more facts than specified by the property, but performance may degrade significantly. CEP server will not shutdown if limit is exceeded.	Integer. Default is 10000.
server.topic.cache.size	Maximum size of the cache (number of objects). This cache is used to cache remote objects. Also see server.topic.cache	Integer. Default is 1101.
server.topic.request.limit	Maximum number of outstanding requests for the topic.	Integer. Default is 10000.
server.publish.force	Force publishing a bypass duplicate check. Reduces network bandwidth utilization and CPU and memory consumption.	Boolean. Default is false.
server.facts.history.maxsize	Server-wide history size for all collected facts. Maintains history for number of samples selected for all facts.	Integer. Default is 0.
server.facts.history.maxtime	Server-wide history time for all collected facts. Maintains history for time-base collection of samples in milliseconds for all facts.	Integer. Default is 0.
server.facts.expire	Server-wide facts expiration in ms.	Integer. Default is 0.
server.facts.log.incr	Enable only incremental logging for facts that are added when logging is enabled.	Boolean. Default is false.
server.facts.expiry.rate	Rate at which expiration of facts is checked, lowering value decreases performance.	Integer. Default is 60000.
server.facts.expiry.debug	Enable/disable fact expiry debug trace.	Boolean. Default is false.

Table 5-11. Server Properties

Property	Description	Values
server.registry.backups	Maximum number of backups for registry files.	Integer. Default is 5.
server.auto.save.timeout	Force auto save after registry change within specified ms.	Integer. Default is 3000ms.
server.system.facts.expire	Set expiration for system facts in ms.	Integer. Default is 0.
server.max.stdout.size	Maximum log size in bytes for stdout and stderr output.	Integer. Default is 500000 bytes.
server.service.facts.logging	For Experts and Managers ONLY. CEP server records all facts produced by the service which can be played back using apfact utility.	Boolean. Default is true.
java.awt.headless	For UNIX ONLY. Enable AWT headless mode to prevent AWT/X11 related activity for server applications which can generate exceptions at runtime.	Boolean. Default is true.

Table 5-12. Grid Properties

Property	Description	Values
server.grid.folder	Location of grid definitions.	String. [AUTOPILOT_HOME]/ naming.grid
server.grid.vote.sample	Wait interval before selecting primary grid server.	Integer. Default is 120000.
server.grid.vote.delay	Wait before voting on a primary after primary failed.	Integer. Default is 15000.
server.grid.dynamic.services	All services in the grid are dynamic – nonpersistent. Removed from directory when hosting server is not available.	Boolean. Default is false.

Table 5-13. JDBC Properties

Property	Description	Values
server.jdbc.oracle.driver	Oracle Driver	oracle.jdbc.driver.OracleDriver
server.jdbc.oracle.url	Oracle URL	jdbc:oracle:thin:@
server.jdbc.sqlserver.driver	SQL Driver	net.sourceforge.jtds.jdbc.Driver
server.jdbc.sqlserver.url	SQL URL	jdbc:jtds:sqlserver://
server.jdbc.sybase.driver	Sybase Driver	net.sourceforge.jtds.jdbc.Driver

Table 5-13. JDBC Properties

Property	Description	Values
server.jdbc.sybase.url	Sybase URL	jdbc:jtds:sybase://
server.jdbc.db2.driver	DB2 UDB Driver	COM.ibm.db2.jdbc.net.DB2Driver
server.jdbc.db2.url	DB2 UDB URL	jdbc:db2://
server.jdbc.hypersonicsql.driver	Hypersonic SQL Driver	org.hsql.jdbcDriver
server.jdbc.hypersonicsql.url	Hypersonic SQL URL	jdbc:HypersonicSQL:hsql://
server.jdbc.jdbc.odbc.driver	Generic JDBC/ODBC Driver	sun.jdbc.odbc.JdbcOdbcDriver
server.jdbc.jdbc.odbc.url	Generic JDBC/ODBC URL	jdbc:odbc:
server.jdbc.informix.driver	Informix Driver	com.informix.jdbc.IfxDriver
server.jdbc.informix.url	Informix URL	jdbc:informix-sqli://
server.jdbc.mysql.driver	MySQL Driver	com.mysql.jdbc.Driver
server.jdbc.mysql.url	MySQL URL	jdbc:mysql://
server.jdbc.derby.driver	Derby Driver	com.ibm.db2.jcc.DB2Driver
server.jdbc.derby.url	Derby URL	jdbc:derby:net://

5.4 Key Performance Properties

Several key performance properties may require customization:

- `server.facts.capacity`
- `server.agent.timeout`
- `server.topic.cache`
- `server.topic.cache.size`

The most important property is `server.facts.capacity`, which determines the maximum number of managed properties that can be monitored. Determine if the CEP server or domain server requires property changes by checking `[MANAGED_NODE]_Facts` as follows:

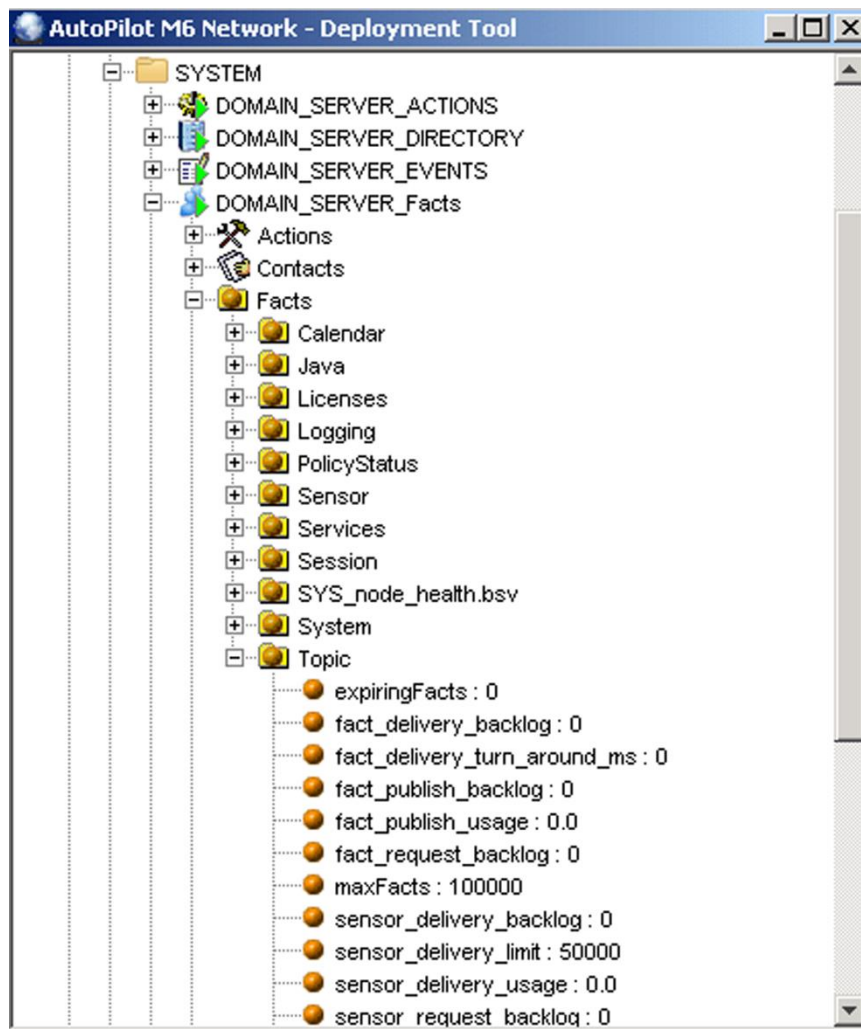


Figure 5-2. CEP Server, Domain Server Facts

If `[MANAGED_NODE]Facts\Topic\totalFacts` is less than `[MANAGED_NODE]_Facts\Topic\maxFacts` then:

`server.facts.capacity` will require modification to: $totalFacts + 1/3 * totalFacts$.

(Example: Where $maxFacts = 700$ and $totalFacts = 600$, increase $totalFacts$ by $totalFacts$ plus $1/3$ or $totalFacts = 800$).

Chapter 6: Troubleshooting Techniques

This chapter describes general troubleshooting techniques for any of the meshIQ Platform Core Services components. Since the platform is a distributed network of servers and services, it could make problem determination somewhat difficult. This section describes error reporting and problem determination facilities.

6.1 Overview

Troubleshooting Facility is categorized as follows:

- **Event Logs:** Event/error logs that are local to each installation.
- **Service Activity:** Allows users to enable traces for any management service within the meshIQ domain for the purpose of problem determination. Traces can be viewed using Event Viewer.
- **Event Viewer:** Centralized Event Viewer allows the user to view, filter, sort events, and errors on any installation, local or remote. Event Viewer can be launched using the User Console.

6.2 Event Logs

Each installation maintains a set of logs for all installed components in directory `[AUTOPILOT_HOME]\logs`. Each log has an `.EVT` file extension. Logs have *Logical* and *Physical Names*:

- **Logical Name:** User known name that is displayed in the Event Viewer title bar and in individual service Logging Properties. Default Logical Names are *SYSTEM*, *UNNAMED*, *POLICIES*, and *SERVICES*. There are system logs and user-defined logs.
- **Physical Name:** physically all logs are located in `[AUTOPILOT_HOME]\logs` directory and have `.EVT` file extension.

Each log file has a physical file name; actual file stored on the file system. Physical name convention is: `[MANAGED_NODE_NAME]_[LOGICAL_NAME]$0|1.EVT` (example: `DOMEGAX_SYSTEM$0.EVT`).

Since logs are circular, Core Services may allocate two logs, for example: `DOMEGAX_SYSTEM$0.EVT`, when log is full, the `$0` log is moved to `DOMEGAX_SYSTEM$1.EVT` and the original file is truncated (deleted and recreated).

6.2.1 System Logs


- **SYSTEM:** System services record to this log.
- **SECURITY_ACTIVITY:** (Domain Server only) Security Service records information on security action. Log is created only when "Service Activity" is set to **true** (on). **DOMAIN_SERVER_SECURITY** service.
- **ACTION_ACTIVITY:** Contains failed/audited user actions.

- **UNNAMED:** Should always be empty. This log is maintained when an event is written to a non-existent log. Any events recorded into this log indicate an internal error and should be reported to [meshIQ technical support](#).

6.2.2 User-defined Logs

- **POLICIES:** All deployed policies record events into this log (unless changed by the authorized user). Log name can be changed/created when modifying individual policy's *Logging Properties*.
- **SERVICES:** Where all management services record failures, errors, and warnings, as well as trace messages, log name can be changed/created when modifying individual service's *Logging Properties*.

Users may change log names by setting *Log name* of the *Logging* properties section of every management service.



All log files are stored as plain text files, viewable using any text editor.

Each event in every .EVT event log is formatted as follows:

- **key:**{time=*timestamp*}{type=*5*}{eid=*id*}{account=*user*}{from=*component*}{event=*message*}

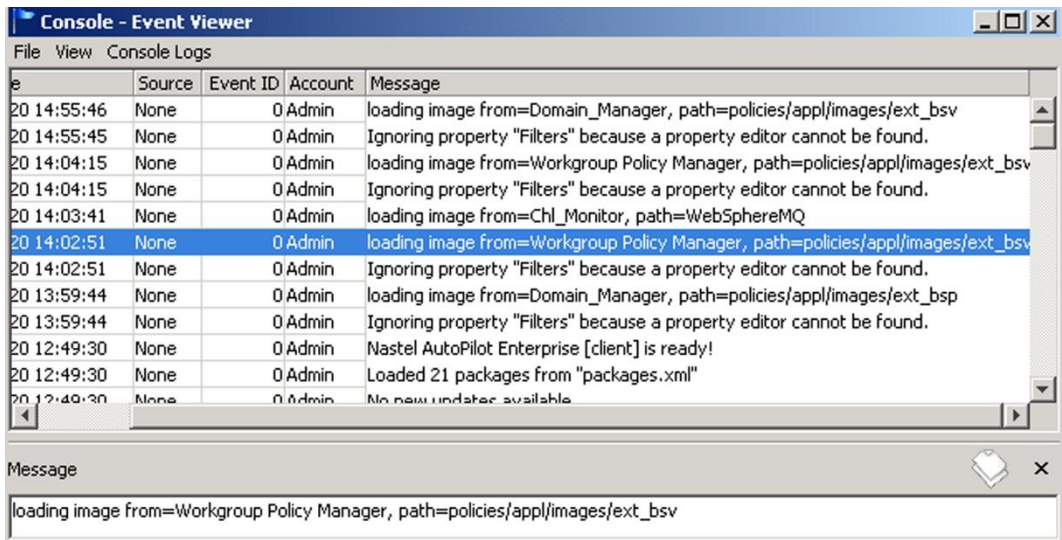


Figure 6-1. Sample Event Log

6.3 Service Activity



CAUTION!

Log Service Activity should be enabled only during problem determination. Leaving Log Service Activity enabled will cause significant performance degradation.

Each management service can be enabled to generate Service Activity or detailed trace information. Enable "Service Activity" on a specific management service as follows:

1. Right-click desired management service (Example: expert, manager, or policy) to open sub-menu.
2. Click **Properties**.
3. Click the **Logging** tab.
4. **Log Name** is user defined. Logs are system-generated if an existing log is available.
5. **Log Service Activity**: Click the disable/enable button to enable log service activity. A check mark will be displayed. Select *Audit* to enable audit events recorded every time any user accesses this service.
6. Click **Apply** to apply the changes to the selected service.

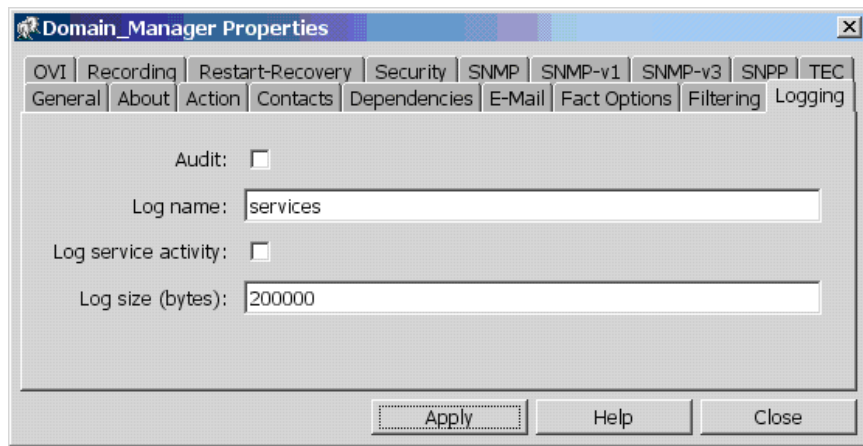


Figure 6-2. Logging, Service Activity



NOTE

When *Log Service Activity* or *Audit* is enabled and applied, logs can be viewed using Event Viewer under the CEP server where service is running.

- Open the event view by right-clicking the subject manager, expert, or policy, the event viewer will be displayed. Open the event file from the file menu either by clicking **Open**, or from the *Domain Logs* menu as shown in the figure below.

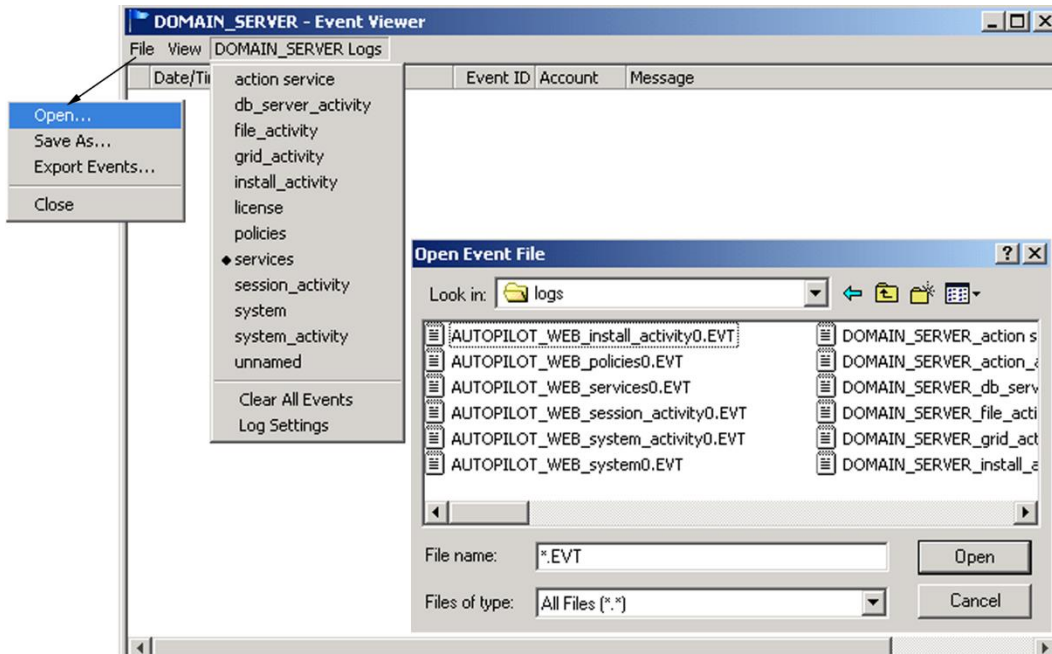


Figure 6-3. Opening Log in Event Viewer

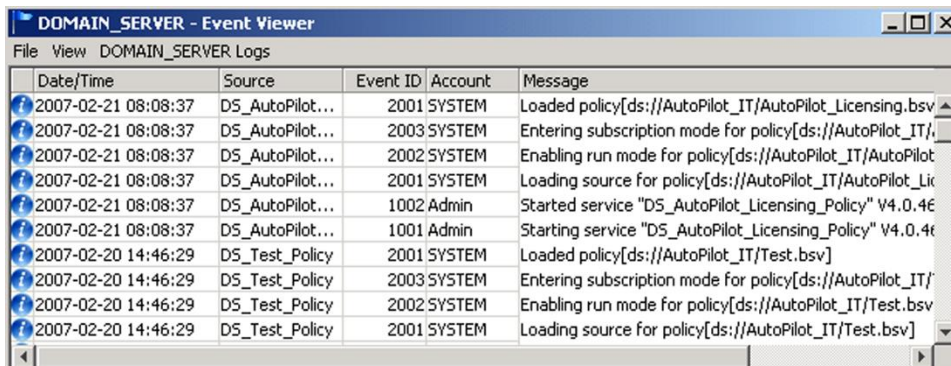


Figure 6-4. Viewing Log in Event Viewer

6.4 Event Viewer

The User Console allows users to launch event viewer for each CEP server, manager, expert, or policy. When event viewer is active, users can view all logs generated under specified CEP servers, including remote CEP servers.

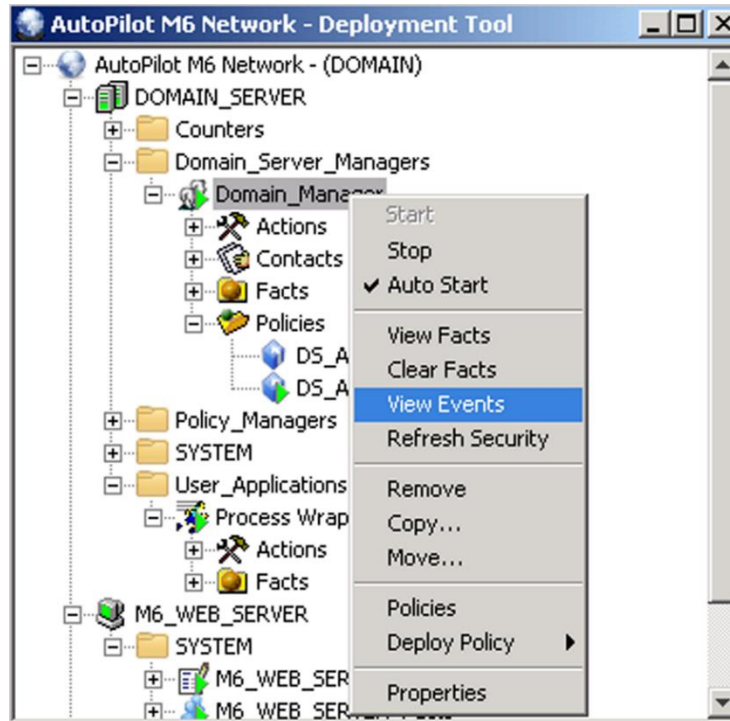


Figure 6-5. Event Viewer for a CEP Server



Event Viewer is also accessible from the User Console (required for troubleshooting User Console activity).

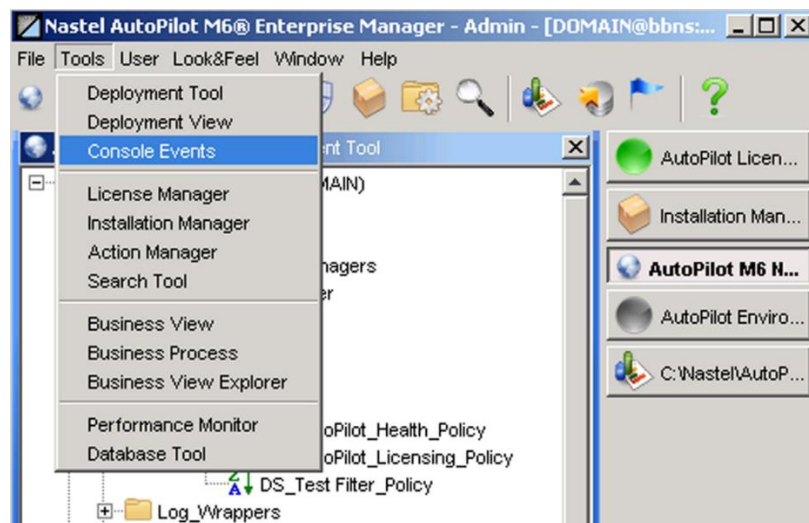
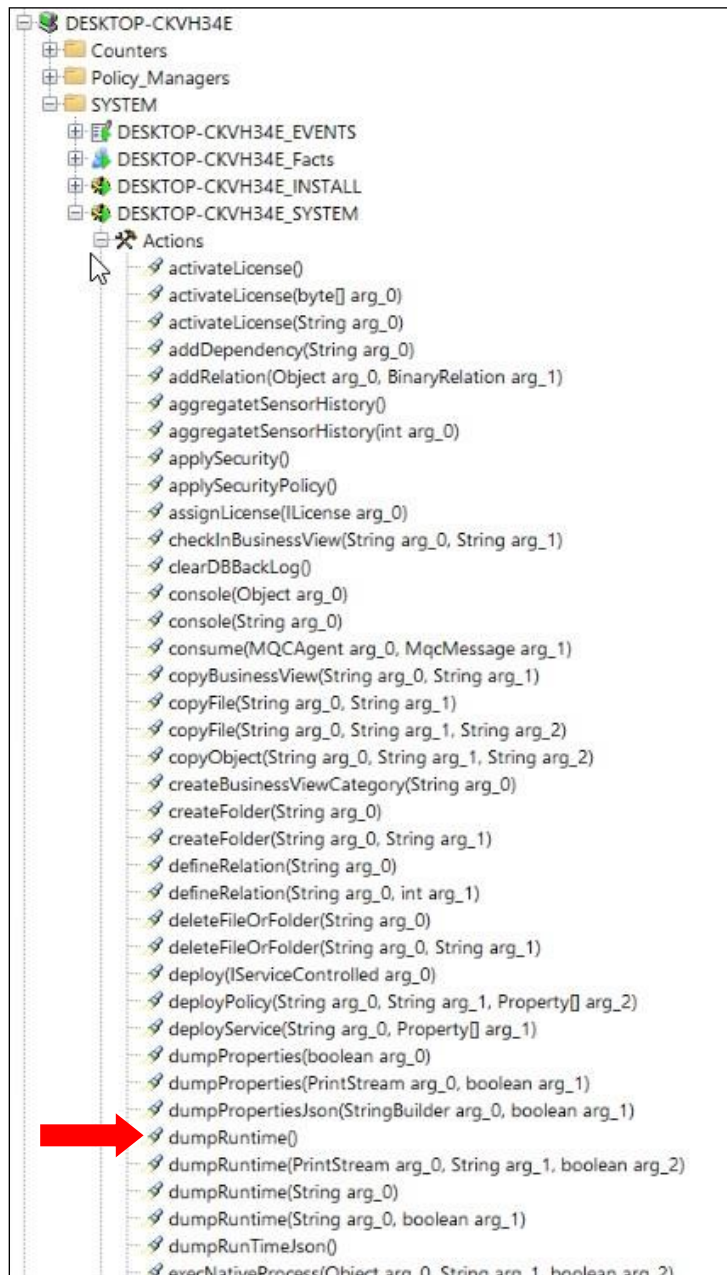


Figure 6-6. Launching Event Viewer from Console Tool Menu

6.5 Generating NRD Files

An NRD file captures the state of the Core Services environment. The file is helpful to technical support as an aid in troubleshooting issues. The file is generated automatically during stress but can be generated on-demand using Enterprise Manager.

To generate an NRD file, right-click **dumpRuntime**, then select **Run Action**.



Right-click **dumpRuntime**, then select **Run Action** to generate NRD file.

Figure 6-7. Generate NRD File

Chapter 7: Troubleshooting Procedures

The following chapter identifies specific problems that may occur while configuring/using Core Services and gives detailed solutions for these problems. If a problem still exists, after reading this chapter, contact meshIQ as follows:

- To contact meshIQ technical support by phone, call 1-800-963-9822 ext. 1. If you are calling from outside the United States, dial 001-631-761-9190.
- To contact meshIQ technical support by email, send a message to mysupport@meshiq.com
- To access the meshIQ automated support system (User ID and Password required), go to <https://mysupport.meshiq.com/>

Contact your local meshIQ Platform Administrator for further information.

Before requesting support, please verify the following:

- The latest service update is installed.
- All the latest versions of your plug-ins are installed.
- All components (for example, CEP Servers, the Domain Server, and the Domain Console) are at the same version and service level.
- The license has been installed and has not expired.
- The CPU license limit has not been reached.
- All services are started (Domain Server, CEP Server, and/or Web Server (if applicable)).

7.1 Problems and Solutions

Problems along with their solutions are grouped into categories as follows:

- Installation
- Business Views
- Facts
- CEP Server

7.1.1 Installation

When running `pkgman ../updates/filename` command - returns a warning that says: `Package database is currently locked. Please try later.`

Delete `*.lock*` file in `[AUTOPILOT_HOME]` directory. This file is created when `pkgman` is killed before completing.

7.1.2 Business Views

Is there a way to turn on and off monitoring of a specific sensor in a Business View?

Core Services has the ability to specify selected sensors within business views to ignore faults and alert conditions during a specified window of time.

1. Right-click sensor to be ignored.
2. In the sub-menu, select **Properties**, the *properties* screen will be displayed.
3. Click *the Ignore* tab to open the ignore properties screen.
4. The default settings have all days selected, with no time specified. This means that no ignore features are active. By deselecting a day, default time of "5 PM" is displayed, but the ignore feature is disabled.
5. Enter the time, for the day(s) you require.
6. Enter the number of minutes you want the sensor to be ignored (example: 12 Hours = 720 minutes, 24 hours = 1440 minutes).
7. Select (check) desired day(s). Sensor activity for the day(s) and time specified will be ignored.
8. Ensure all other days are disabled to prevent inadvertent disruption of sensor data.
9. Click **Apply** when all day and time settings are set. All information for specified period will be ignored.

Is there a way to view images in the description field of a business view?

The *Description* field is not a fully enabled HTML browser. The recommended approach is to use a link to your pages that contains HTML content including dynamic content, forms, etc. Since business views can also be displayed on the web, it is best to author all HTML pages using a different tool and post them on the web server and then link from a business view description to those pages. The `.../images/cs.gif` convention would not work since web servers would not be able to locate these images when business views are loaded from the web. Therefore, it is recommended to use off-line web pages and link to them from the *description* field.

How does an administrator of AutoPilot unlock BSV currently locked by someone else?

A BSV can be unlocked by using the Business View Explorer.

1. From the User Console toolbar, click **Tools > Business View Explorer**.
2. Right-click locked BSV and click **Check-In & Unlock**.

7.1.3 Facts

Is there a way for user to selectively batch remove published facts from the User Console?

1. Right-click on desired expert.
2. In the sub-menu, select **Properties**, the *Properties* screen will be displayed.
3. Click the *Fact Options* tab.
4. In the *Expire facts(ms)* field, enter the desired value in milliseconds for when you want the old fact to be cleared within that expert.
5. Click **Apply**.

Is there a way to publish facts to the meshIQ Platform from your java applications?

1. Import the following package:
`com.nastel.nfc.util.*`
2. Declare the following:

```
String M6_HOST = "localhost"; // This is host or IP of the CEP server.  
int M6_PORT = 6000; // This is port on which a UDP listener is  
listening.  
RecordFact sendFact = new RecordFact();  
RecordFact class is located in %AUTOPILOT_HOME%\lib\nfc.jar.
```
3. Use the reportFact method in your code to send facts like the following:

```
sendFact.reportFact("MyApp\\State=true", M6_HOST, M6_PORT);
```

This interface can also be leveraged by non-JAVA applications as follows:

- The interface can be called from the command line and used by C programs using a system function.
- You can also publish a UDP/TCP string to the port in the format accepted by apfact. (See usage display for `[AUTOPILOT_HOME]\bin\apfact`). If you start a user application using a Process wrapper expert, the output produced from the application will automatically create facts. The format of the output must match the format accepted by apfact.

7.1.4 CEP Server

CEP Server will not start and generates an error log.

Some possibilities may include:

1. Inconsistency of Service Updates. Ensure the latest service pack is installed for all environments.

-
2. Incorrect Host Name and/or Port. Modify global.properties file to ensure the correct Host Name and Port are being used.

7.2 FAQs

7.2.1 How to Stop/Start Core Services

Domain Server:

Start: `[AUTOPILOT_HOME]/naming`, type: **nohup ./ ATPNAMES &**

Stop: `ps -aef | grep ATPNAMES` and kill naming process

OR `[AUTOPILOT_HOME]/bin`, type: **apnet -domain localhost stop DOMAIN_SERVER_SYSTEM**

CEP Server:

Start: `[AUTOPILOT_HOME]/localhost`, type: **nohup ./ATPNODE &**

Stop: `ps -aef | grep ATPNODE` and kill localhost process.

OR `[AUTOPILOT_HOME]/bin`, type: **apnet -domain localhost stop <NODENAME>_SYSTEM**

Web Server:

Start: `[AUTOPILOT_HOME]/apache-tomcat/bin`, type: **catalina.sh run &**

Stop: `ps -aef | grep tomcat` and kill tomcat process.


OR `[AUTOPILOT_HOME]/apache-tomcat/bin`, type: **/shutdown.sh**

For Tworks MQ probe:

Start: `nsqtacon -console [options based on TD environment]`

Stop: `nsqtacmd -stop`

7.2.2 What does the icon (gray ball with green refresh arrows) represent?

The icon  means incomplete, meaning sensor does not have enough info to evaluate itself. Usually happens if one or more underlying facts are not available. When this occurs, the sensor goes into a halt state until all facts can be resolved. This can also occur if the rule itself has a problem causing the sensor to go into a halt state.

Appendix A: References

A.1 Documentation

Table A-1. Documentation	
Document Number (or higher)	Title
CS-INS11.000	<i>Core Services Installation Guide</i>
M6/MQ 10.002.1	<i>AutoPilot® M6 Plug-in for IBM MQ Installation and User's Guide</i>

A.2 IBM Documentation

IBM MQ 9.3: <https://www.ibm.com/docs/en/ibm-mq/9.3>

MQSC Command Reference: <https://www.ibm.com/docs/en/ibm-mq/9.4?topic=reference-mqsc-commands>

A.3 HP OpenView Documentation

http://ovweb.external.hp.com/lpe/doc_serv/

A.4 Java™ 2 J2SE™ for HP-UX Information Library

<http://www.hp.com/products1/unix/java/infolibrary/index.html>

<http://developer.java.sun.com/developer/technicalArticles/Servlets/corba/>

A.5 Apache Tomcat Documentation References

<https://tomcat.apache.org/tomcat-10.1-doc/index.html>

A.6 Oracle Online Documentation

<http://otn.oracle.com/documentation/content.html>

A.7 Tru64 UNIX Online Documentation and References

http://h30097.www3.hp.com/docs/pub_page/doc_list.html

Appendix B: Conventions

B.1 Typographical Conventions

Table B-1. Typographical Conventions	
Convention	Description
Blue/Underlined	Used to identify links to referenced material or websites. Example: meshIQ.com
Bold Print	Used to identify topical headings and toggles or buttons used in procedural steps. Example: Click EXIT .
<i>Italic Print</i>	Used to place emphasis on titles, menus, screen names, or other categories.
Monospaced Bold	Used to identify keystrokes/data entries, file names, directory names, etc.
<i>Monospaced Italic</i>	Used to identify variables in an address location. Example: <code>[AUTOPILOT_HOME]\documents</code> , where the portion of the address in the brackets [] is a variable.
Monospaced Text	Used to identify addresses, commands, scripts, etc.
Normal Text	Typically used for general text throughout the document.
Table Text	Table text is generally a smaller size to conserve space. 10-, 9-, and 8-point type is used in tables throughout the meshIQ Platform family of documents.

Appendix C: Command Reference

This Appendix describes the Core Services command line interface, usage, and options.

C.1 PKGMAN – Product Maintenance

pkgman is a product maintenance utility. It allows users to:

- Display information about all installed components and libraries
- Repair damaged installations
- Verify installed packages
- Install plug-ins, service packs and patches from a local file system or URL location.



All output from **pkgman** is stored in `[AUTOPILOT_HOME]\logs\pkgman.log` in addition to console output, even if executed in GUI mode.

Table C-1. PKGMAN Options

Option	Description
-gui	Runs in GUI mode, opens the Package Manager in GUI mode (default console). (See below.)
-about	Displays product information (see below)
-info	Displays installed product information. Displays installed product component information. (See below.)
-libinfo	Displays library information. (See below.)
-verify	Verifies installed package. Displays the details of the last installation package. <code>pkgman -verify [package name]</code>
-repair	Repairs the installation of the package identified. <code>pkgman -repair [package name]</code>
-uninstall	Uninstalls an existing package. <code>pkgman -uninstall [package name]</code>
-reinstall	Reinstalls from a package file. <code>pkgman -reinstall [package name]</code>
-help	Prints this message.

GUI Mode

The **pkgman** will run as a GUI application when **-gui** command line option is used. It can also be invoked by running **apman** at the command prompt or (Windows only) accessed from the Core Services start menu, under the *Product Maintenance* link.

Product Information

Displays the product information when the **-about** option is executed (example: `pkgman -about`).

Installed Product Information

In the sample, the base identified is defined along with the build version and date. Details are displayed by using the **-info** option (`pkgman -info`).

Library Information

The information about installed java packages (jar files) is displayed when using **-libinfo** option (`pkgman -libinfo`).

MIME File Type Recognition

Mime types are searched in the following order:

- The file `.mime.types` in the user's home directory.
- The file `[java.home]/lib/mime.types`.
- The file or resources named `[AUTOPILOT_HOME]/bin/META-INF/mime.types`.
- The file or resource named `META-INF/mimetypes.default` which is located in `[AUTOPILOT_HOME]/lib/activation.jar` file.

MIME Types File Format

- # comments begin with a "#"
- # the format is `[mime type][space separated file extensions]`
- for example: `text/plain txt text TXT`
- # this would map `file.txt`, `file.text`, and `file.TXT` to the mime type "text/plain"

C.2 APNET – Control Utility

apnet allows users to control services from a command line, locally or remotely. **apnet** requires that the domain server be up and running and available on the network. Users don't have to specify the location of the domain server, user, and password if **apnet** is executed on the same server as the domain server.

Table C-2. APNET Options

Option	Description
-useglb	Use settings supplied in <code>[AUTOPILOT_HOME]/global.properties</code> (default is false)
-dsname	<code>[domain server name]</code> domain server host name (default is DOMAIN_SERVER)
-domain	<code>[domain server host name]</code> domain server host name (default is localhost)
-port	<code>[domain server port]</code> domain server port (default is 2325)
-user	<code>[user name]</code> user name
-password	<code>[password]</code> user password
-relation	<code>[relation_name]</code> relation name (default is DEPENDS_ON)
-pair	<code>[ordered_pair]</code> ordered pair (default is: (*,*) -- all)
Service	Service or policy name. Policy is specified as policy_name:manager_name
autostart	Sets the auto-start for the designated service. The service will run whenever the CEP server is active.
start	Manually starts the designated service.
stop	Manually stops the designated service.
disable	Disables the designated service. Service required manual start once disabled.
remove	Removes the designated service.
save	Saves current configurations on the specified CEP server. CEP server is specified as follows: <code>[NODE_NAME]_SYSTEM</code> .
query	Queries and displays service registration attributes.
clear	Clears all facts for a specified service. Executed asynchronously, inline and serialized with published facts to preserve order.
get	Queries facts and value pairs for a given pattern.
lookup	Looks up any registered object in the domain server (service and non-service).
change_pwd	Resets account password. apnet must run on the same box as the domain server or user must use Admin User ID/password to reset account password.
reset-pwd Admin	Resets account password for the Administrator.

Table C-2. APNET Options

Option	Description
ignore	<p>Deactivates object monitoring for a limited time (by toggling the fact state to ignore). Useful when users do not want notifications for specific objects. Toggle the fact state as follows: <code>apnet.exe ignore <ServiceName> fact</code></p> <p>To reenable object monitoring, run the command again for the particular object. Please note that the ServiceName should be in double quotes if there are spaces: <code>apnet.exe ignore "Test Counter" counter</code></p>

Examples:

```

apnet query DOMAIN_SERVER_SYSTEM
apnet stop Domain_Server_Manager
apnet clear Domain_Server_Manager
apnet -domain localhost -port 2325 save DOMAIN_SERVER_SYSTEM
apnet lookup DOMAIN_SERVER
apnet ignore Que_Monitor MQM\KITE\QM\BBB.LQ\CURDEPTH
apnet ignore Que_Monitor MQM\KITE\QM\BBB.LQ\*

```

apnet also executes methods used within services such as experts, managers, and policies.

Example (when executing JMX MBean methods):

```
apnet invoke JMXAgent.invoke(Domain,MBeanName,mbean_method.String[]=null)
```

where:

JMXAgent is the name of the service.

invoke is the name of the JMXAgent method.

Service follows the following format when used with the `invoke` command:

```
ServiceName.method(type=value,...)
```

where:

type can be a *String*, *Integer*, *Float*, *Double*, *Boolean*, *String[]*, *Integer[]*, *Float[]*, *Double[]*, or *Boolean[]*

value is the value of the variable to be passed to the calling method.



NOTE

Spaces are not allowed unless part of the value.

Array values are specified in the following format [`'val1'val2'...'valN'`] with no spaces between values.

Example: `String[]=['This'is'a'test']` is a list of four strings.

C.3 APFACT – Fact Publisher

apfact utility publishes facts to a process wrapper, allowing users to instrument scripts and user applications without coding. The utility uses UDP protocol to send facts to the Process Wrapper. The Process Wrapper expert must be deployed and configured to listen on a designated UDP port. That port should be specified on the command line of **apfact**. It is recommended to use **apfact** and Process Wrapper on the same machine, due to limitations of UDP protocol. **apfact** and Process Wrapper should run on the local area network in close proximity (preferably on the same host) and should not be separated by routers (many routers and firewalls block UDP traffic).

Table C-4. APFACT Options

Option	Description
-nobanner	Do not show banner.
-verbose	Enables verbose mode.
-host	Host name of the CEP server (default=localhost) <code>apfact -host [host_name]</code>
-protocol udp tcp	Protocol used to publish facts (default=udp)
-port	Listener port of the Process Wrapper Expert (default=6000) <code>apfact -port [port_number]</code>
-file	Input file containing a batch of facts <code>apfact -file [file_name]</code>
-factor factor_value	Acceleration factor or multiplier by which the time delays in the batch file are multiplied. For example, a .5 factor would cause half the time delays. The combination of factor and repeat allows the same batch file to be used to simulate varying fact generation rate and volume conditions.
-repeat repeat_count	Number of times to repeat batch (default=1)
-obj	Specifies common fact prefix followed by individual metrics.
-filter filter_string	Facts filter applied to batch (default=*)
-fact	<code>fact1=value1[desc][act1:op1+act2=op2..],..., factN=valueN</code>

Example apfacts Sent from the Command Line

- ```
apfact -port 6010 -filter "CPU*" -file input.txt
```

This example specifies a fact batch file is to be used. The filter qualifier specifies that only facts that start with fact category "CPU" should be extracted from the batch file.
- ```
apfact -port 6010 -fact "CPU\IDLE=50"
```

CPU is the fact category and IDLE is an integer fact within the category that will appear in the Enterprise Manager nested under CPU.

3. `apfact -port 6015 -fact "CPU\IDLE=10[CPU Idle percent],CPU\BUSY=90"`
Similar to example 2, but a description is included following the IDLE integer fact value. The description will appear in the fact properties. The additional integer fact BUSY will appear nested under CPU and aligned with IDLE.
4. `apfact -host 123.12.34.15 -port 6010 -fact "CPU\IDLE=10[CPU Idle]"`
Similar to the IDLE fact of example 3, but a specific host IP address versus the default localhost is specified.
5. `apfact -fact "SERVER\Running=false[server state]
[Start:START.BAT+Stop:STOP.BAT]"`

Defining Facts in a File

To define a batch of facts in a file, use the `-file` option and provide a fact file in the following format:

```
time1_in_ms fact1  
time2_in_ms fact2
```

Example:

```
100 CPU\Idle=10  
10 CPU\Busy=90,Swap Time=50
```

The input file is processed line-by-line. `apfact` waits for `time_in_ms` (milliseconds) before publishing the fact, which must be specified in the format described by the `-fact` option. Each fact may require double quotes around it. Examples:

1. `apfact -port 6010 -factor 0.1 -file input.txt`
Read and send the facts defined in file `input.txt`, multiplying the specified time delays by a factor of `.1`. For the example fact batch file shown above, the IDLE fact will be sent every 10 seconds instead of every 100 seconds, and the BUSY fact will be sent with a delay of 1 second instead of 10 seconds.
2. `apfact -port 6010 -repeat 10 -file input.txt`
Repeat 10 times the reading and sending of facts defined in file `input.txt`.
3. `apfact -port 6010 -filter "CPU*" -file input.txt`
Read and send the facts in fact file `input.txt`, filtering to select only facts that start with fact category CPU.

C.4 ATPNODE – CEP Server

ATPNODE starts CEP server component and normally runs in the background on all Core Services hosts. **ATPNODE** can also be stopped/started from Windows Services. **ATPNODE** logs events and errors into `[AUTOPILOT_HOME]\logs` directory. All logs have the name of the CEP server as part of the log name.

Table C-5. ATPNODE Options

{Option} Option_Value	Description
-w [work_dir]	Identifies the work directory where property files are located (default=.) (example: -w [AUTOPILOT_HOME]\localhost)
-p [prop_file]	Fully qualified server property filename. Identifies a user-defined property file Default is [AUTOPILOT_HOME]\localhost\.\node.properties
-d [{domain} dsURL]	Designates the domain server URL (bbns://server:port) -domain bbns://Domain_Server:2325
-b [daemon]	Flag that the node will run in background (default is true) [AUTOPILOT_HOME]\localhost\ATPNODE -daemon
-c [console]	Flag that the node will run in foreground (default=false) [AUTOPILOT_HOME]\localhost\ATPNODE -console
-s [safe]	Starts node in safe mode; all services are disabled (default is false) [AUTOPILOT_HOME]\localhost\ATPNODE -safe
-x [debug]	Prints out all java properties at startup.
-r [{registry} regx_file]	Fully qualified service registry XML file name. Allows switching active service directory. Default is [AUTOPILOT_HOME]\localhost\.\registry.xml
-l [logfacts]	Records all facts for all services that have the <i>Recording</i> option enabled. The file remains open until the server stops. However, will not record if property server.facts.jdbc.driver = com.mysql.jdbc.driver is specified. [AUTOPILOT_HOME]/localhost/[server_name]_Facts.fct
-f [profile]	Enables sensor profiler globally for computation of sensor profiling metrics (default=disabled)
-k [stop]	Stops services on CEP server gracefully.
-x [debug]	Enables debug mode and printing of stack traces/properties (default=false)
-a [strong]	Enables strong authentication; per request (default=false)
-e [export]	For the domain server, exports account information into a text file.
-z [import]	For the domain server, imports accounts from an XML export file.
-n [noupdate]	Upon start of program, will not obtain and install software updates.
-j [{join} grid_name]	Joins server to the specified grid.

Table C-5. ATPNODE Options

{Option} Option_Value	Description
-i [{priority} number]	Specifies grid priority. Used when selecting primary servers (default=0)
-o [{out.log} log_dir]	Overrides the default log directory (default=[install_dir]/logs)
-h [help] -? -usage	Displays all command line options.

ATPNODE -console:

Use the `-console` option to run the node in the foreground.

(Example: `[AUTOPILOT_HOME]\localhost\ATPNODE -console`)

ATPNODE -safe:

Use the `-safe` option to start a node in safe mode, where all services are in stop mode. Used when troubleshooting. (Example: `[AUTOPILOT_HOME]\localhost\ATPNODE -console -safe`)

ATPNODE -debug:

Use the `-debug` option to print all Java properties when starting the node. Used when debugging or troubleshooting. (Example: `[AUTOPILOT_HOME]\localhost\ATPNODE -console -debug`)

ATPNODE -r regx_file:

Use the `-r regx_file` option to switch CEP server active service directory. (Example: `[AUTOPILOT_HOME]\localhost\ATPNODE -console -r registry.xml`)

C.5 ATPNAMES – Domain Server

ATPNAMES starts the meshIQ domain server component. **ATPNAMES** normally runs in the background on all meshIQ Platform Core Services nodes. The CEP server can be started and stopped from Windows Services. **ATPNAMES** logs events and errors into `[AUTOPILOT_HOME]\logs` directory. All logs have the name of the domain server as part of the log name. (Refer to [Chapter 3](#) for additional information.)

Table C-6. ATPNAMES Options

{Option} Option_Value	Description
-w [work_dir]	Identifies the work directory where property files are located (def=.) (example: <code>-w [AUTOPILOT_HOME]\naming</code>)
-p [prop_file]	Identifies a user-defined property file (default is node.properties) (example: <code>-p node.properties</code>)
-d [{domain} dsURL]	Designates the domain server URL (bbns://server:port) <code>-domain bbns://Domain_Server:2325</code>

Table C-6. ATPNAMES Options

{Option} Option_Value	Description
-b [daemon]	Flag that the node will run in background (default=true) [AUTOPILOT_HOME]\naming\ATPNAMES -daemon
-c [console]	Flag that the node will run in foreground (default=false) [AUTOPILOT_HOME]\naming\ATPNAMES -console
-s [safe]	Starts node in safe mode; all services are disabled (default=is false) [AUTOPILOT_HOME]\naming\ATPNODE -safe
-x [debug]	Prints out all java properties at startup.
-r [{registry} regx_file]	Allows switching active service directory (default=registry.xml)
-l [logfacts]	Records all facts for all services that have the <i>Recording</i> option enabled. The file remains open until the server stops. However, will not record if property server.facts.jdbc.driver = com.mysql.jdbc.driver is specified. [AUTOPILOT_HOME]\naming\[server_name]_Facts.fct
-f [profile]	Enables sensor profiler globally for computation of sensor profiling metrics (default=disabled)
-k [stop]	Stops services on CEP server gracefully.
-x [debug]	Enables debug mode and printing of stack traces/properties (default=false)
-a [strong]	Enables strong authentication; per request (default=false)
-e [export]	For the domain server, exports account information into a text file
-z [import]	For the domain server, imports accounts from an XML export file
-n [nouupdate]	Upon start of program, will not obtain and install software updates
-j [{join} grid_name]	Joins server to the specified grid
-i [{priority} number]	Specifies grid priority. Used when selecting primary servers (default=0)
-o [{out.log} log_dir]	Overrides the default log directory (default=[install_dir]/logs)
-h [help] -? -usage	Displays all command line options

ATPNAMES -console:

Use the `-console` option to run the domain server in the foreground.

(Example: `[AUTOPILOT_HOME]\naming\ATPNAMES -console`)

ATPNAMES -safe:

Use the `-safe` option to start a domain server in safe mode. Used when troubleshooting.

(Example: `[AUTOPILOT_HOME]\naming\ATPNAMES -console -safe`)

ATPNAMES -debug:

Use the `-debug` option to print all Java properties when starting the domain server.

(Example: `[AUTOPILOT_HOME]\naming\ATPNAMES -console -debug`)

Appendix D: Best Practices

D.1 Naming Conventions

For optimum performance while using meshIQ Platform Core Services, use the following recommended naming conventions.

- Avoid using blanks in service, policy, and user definitions.
- Avoid using special characters such as \$ * # ; : etc.
- Experts should be named as <prefix>_Monitor
- Managers should be named as <prefix>_Manager

Under the Manager *General* tab in *Properties*, select *Naming Convention* and enable *Enforce Naming Convention* so that all policies will have the same naming convention automatically applied by the manager.

- Policies should be named as "<prefix>_Policy"
- Use service *Context* property to categorize services by category and usage or purpose.

Under the Services *General* tab in *Properties*, enter category and usage or purpose in *Context* field.

D.2 Guidelines for Building Policies

D.2.1 Modeling

- Avoid hard-coded names and use Global environmental variables such as Java variables and CEP server variables (node.properties) and/or Policy environmental variables.
- Avoid duplication by promoting modularity and reuse.
 - Define and enforce naming conventions
 - Focus on creating modular, reusable business views
 - Create complex policies by combining simple policies.
- Avoid large and complex policies.
 - Build small, simple, self-contained policies
 - Link policies together – avoid using copy and paste
 - Reference policies by reusing existing logic.
- Perform complex event processing using Business Views
 - A Business View is an event processor and should be used as one.

D.2.2 Deployment

- Deploy policies close to their source by using
 - Same CEP server as the origin for most of the facts (performance)
 - Same server as the origin for most of the facts
- Always consider security
 - Who do you want the policy to be accessible to? Should it be available on the web? If so, consider the correct permission model
 - Edit policy security properties to set permissions and ownership.
- Each CEP server should:
 - Be self-sufficient and independent of other CEP servers
 - Contain all the monitors and policies for monitoring.
- Make use of multiple manager instances for:
 - Improving performance
 - Improving response time
 - Group by policy category or purpose

D.2.3 Usage - Reducing Rule and Processing Delays

- Avoid using blocking scripts within sensors because they will:
 - Cause rule execution delays
 - Decrease effective rule per second processing speed
- Avoid excessive dynamic sensor creation and deletion during policy execution.
 - Structure rules so that dynamic sensors are created only on error conditions and removed when condition is resolved.
- If possible, avoid sensors that use multiple facts for evaluation.
 - The more facts included in the sensor, the more processing required during rule execution.

D.3 Key Performance Indicators

D.3.1 System Calendar

The following metrics are published under `<NODE_NAME>_Facts\Calendar`.

Table D-1. System Calendar		
Calendar	Type	Description
day_of_month	Integer	Current day count within the current month.
day_of_week	Integer	Current day count within the current week.
day_of_week_in_month	Integer	Current day of week count within the current month.
day_of_year	Integer	Current day count within the current year.
month	Integer	Month count within current year. For example: 6 corresponds to the month of June.
time_zone	String	Current time zone (EDT, CDT, GMT, etc.)
week_of_month	Integer	Current week count within the current month.
week_of_year	Integer	Current week count within the current year.
year	Integer	Current year.

D.3.2 Java Memory

The following metrics are published under `<NODE_NAME>_Facts\Java`.

Table D-2. Java Memory

Java	Type	Description
absolute_memory_utilization	Float	Ratio of memory used to maximum heap size. Multiply this number by 100 to get percentage of used memory.
free memory	Long	Number of bytes available memory based on the currently allocated heap size (not based on the maximum heap size).
heap_size_max	Long	Maximum memory that can be claimed by the server. If consumed 100%, then no more memory can be allocated, and CEP server will fail.
heap_utilization	Float	Current heap utilization. Multiply by 100 to percentage usage. If reaches 1.0 (100%), CEP server will no longer be able to claim more memory and will fail.
memory_growth_rate_sec	Float	Number of bytes claimed per second. If the number is positive, memory is growing. If the number is negative, memory is shrinking. The value is computed based on the previous sample (every 30 seconds).
memory_utilization	Float	Current heap utilization based on the currently allocated heap size which could be less than or equal to the maximum heap size.
overhead_usage_bytes	Float	Number of bytes used per functional unit which is defined as <i>fact + sensor</i> . This number can be used to estimate the required memory for a given number of facts + sensors. For example: if <code>over_head_usage_bytes = 6000</code> , then to support 10000 facts + 10000 sensors, the user will need $6000 * (2000)$ bytes.
total_memory	Long	Total number of bytes allocated. This number will always be less than or equal to <code>heap_size_max</code> . This number will usually be close to the number of bytes claimed from the operating system.
total_thread_groups	Integer	Total number of allocated thread groups by the CEP server. Each group may hold 0 or more threads.
total_user_threads	Integer	Total number of threads running. Large number of threads may reduce CEP server performance due to thread management overhead.
used_memory	Long	Current number of bytes used by the CEP server within the Java Virtual Machine. This number does not indicate the amount of memory claimed from the operating system.

D.3.3 Database Logging

The following metrics are published under `<NODE_NAME>_Facts\Logging`:

Table D-3. Database Logging		
Logging	Type	Description
db_arrival_rate_per_sec	Float	Number of SQL insert/update requests coming per second. Requests usually come from sensors which are enabled for logging.
db_delivery_rate_per_sec	Float	Number of SQL insert/update requests processed by the database.
db_flow	Boolean	Indicates whether database logging is enabled or disabled. The flow is disabled when db_queue_size reaches db_max_queue_limit to prevent memory exhaustion.
db_last_error	String	Last error encountered during SQL logging.
db_latency_ms	Long	Number of milliseconds that an SQL request sits on a queue before being processed by the database.
db_max_queue_limit	Integer	Maximum queue size for holding all outstanding SQL operations.
db_queue_size	Integer	Current number of outstanding SQL requests.
db_total_arrived	Long	Total number of issued SQL requests.
db_total_dropped	Long	Total number of dropped SQL requests. SQL requests are dropped when db_flow is false.
db_total_errors	Long	Total number of SQL related errors encountered during logging.
db_total_processed	Long	Total number of successfully processed SQL requests.

D.3.4 Sensor Performance

The following metrics are published under `<NODE_NAME>_Facts\Sensor\Performance`:

Table D-4. Sensor Performance		
Sensor/Performance	Type	Description
absolute_rate_rules_per_sec	Float	Number of rules processed per second since start of the CEP server.
absolute_rate_sensors_per_sec	Float	Number of sensors processed per second since start of the CEP server.
last_sensor_exec_time_ms	Float	Time it took to execute last sensor (in milliseconds)
max_rule_exec_time_ms	Float	Longest time it took to execute a rule.
max_rule_hotspot	String	Name of rule associated with max_rule_exec_time_ms.
max_sensor_hotspot	String	Name of the sensor associated with max_rule_hotspot.
min_sensor_exec_time_ms	Float	Minimum time it took to execute a sensor.
rate_rules_per_sec	Float	Number of rules processed per second, based on the total time spent processing rules.
rate_sensors_per_sec	Float	Number of sensors processed per second, based on the total time spent processing sensors.
sensor_busy_percent	Float	Percentage of time used to process sensors.
sensor_idle_percent	Float	Percentage of sensors spent in idle state (not processing rules).
sensor_turn_around_time_ms	Long	Time it takes to process a given metric by one or more sensors. The larger the value the longer it takes from the point of metric discovery to it being processed by the sensor and showing up within business views. Growth of this value usually means that the rule engine is in overload and cannot keep up with the load.
total_processed_rules	Long	Total number of processed rules since last reset.
total_processed_sensors	Long	Total number of processed sensors since last reset.
total_sensor_time_ms	Long	Total time spent processing sensors since last reset (in milliseconds).

D.3.5 Sensor Runtime

The following metrics are published under `<NODE_NAME>_Facts\Sensor\Runtime`:

Table D-5. Sensor Runtime

Sensor/Runtime	Type	Description
sensor_arrival_rate_per_sec	Float	Number of sensor events queued up for processing per second.
sensor_delivery_rate_per_sec	Float	Number of sensor events processed per second.
sensor_dropped_events	Long	Total number of dropped sensor events due to overload.
sensor_failed_actions	Long	Total number of failed user actions (launched by sensors)
sensor_failed_notifications	Long	Total number of failed notifications (SMTP)
sensor_table_size	Long	Total number of outstanding sensor requests.
sensor_total_action_threads	Integer	Total number of threads allocated to execute user actions.
sensor_total_expiring_threads	Integer	Total number of threads associated with expiring sensors (dynamic child sensors)
sensor_total_notifications	Long	Total number of successfully sent notifications (SMTP)
sensor_total_running	Long	Total number of actively running sensors, including user-defined and dynamically created sensors.
sensor_worker_active_threads	Integer	Total number of threads allocated to process all sensors.
sensor_worker_thread_pool_size	Integer	Total number of threads within the thread pool designated for processing rules.

D.3.6 Session

The following metrics are published under `<NODE_NAME>_Facts\Session`:

Table D-6. Session		
Session	Type	Description
avg_pipe_arrival_rate_per_sec	Float	Average rate at which objects are published for delivery to remote clients (subscribers)
avg_pipe_delivery_rate_per_sec	Float	Average rate at which objects are delivered to remote clients (subscribers)
avg_pipe_usage	Float	Average communication usage across all connections. Multiply by 100 to get percentage value.
max_resp_time	Long	Maximum response time for remotely executed actions.
max_send_queue_limit	Long	Maximum queue size of holding outbound requests for each communication session (between client/server).
total_bytes_sent	Long	Total number of bytes sent across all connections.
total_objects_sent	Long	Total number of objects sent across all connections.
total_objects_dropped	Long	Total number of objects dropped due to communication overload, meaning the remote end unable to keep up with consumption of data.
total_objects_recvd	Long	Total number of objects received across all connections.
total_pipes	Integer	Total number of active communications between client and server.
total_pipes_disabled	Integer	Total number of pipes disabled due to communication overload. Disabled pipes will drop objects.
total_pipes_full	Integer	Total number of connections in full state – 100% capacity utilization – their max queues are utilized to 100%.
total_pipes_idle	Integer	Total number of pipes idle.
total_requests	Long	Total number of issued requests (commands) to remote locations.
total_resp_timeouts	Long	Total number of timed out requests.
total_send_queue_depth	Long	Total number of objects queued up for outbound delivery.
total_user_timeouts	Long	Total number of timed out user-initiated requests.

D.3.7 Topic

The following metrics are published under `<NODE_NAME>_Facts\Topic`:

Table D-7. Topic		
Topic	Type	Description
expiringFacts	Long	Total number of expiring facts; facts marked for expiration.
fact_delivery_backlog	Long	Total number of facts waiting to be published to corresponding subscribers. Growing number indicates that the publishing is at much higher rate than can be consumed by subscribers.
fact_delivery_turn_around_time_ms	Long	Time it takes to deliver a given fact to a subscriber. Growing number indicates longer latency and delayed processing by the rule engine.
fact_publish_backlog	Long	Number of outstanding facts waiting to be published.
fact_request_backlog	Long	Total number of outstanding requests for fact lookup.
sensor_delivery_backlog	Long	Total number of sensors waiting to be processed by the rule engine. Growing number indicates that the rule engine is overloaded and may not process information in a timely fashion.
sensor_delivery_limit	Long	Maximum number of allowed sensors that can be in waiting state before being dropped. Once the sensor_delivery_backlog exceeds this limit, some sensor activity will be dropped to prevent CEP server from consuming large amounts of memory and failing.
sensor_request_backlog	Long	Total number of sensors waiting for metric lookup.
totalFacts	Long	Total number of published facts.
totalFilters	Long	Total number of unique filters used by all created sensors.
totalSubscribers	Long	Total number of subscribers, which includes sensors as well as remote subscribers.
utilization	Float	Fact utilization $\text{totalFacts}/\text{maxFacts}$. Multiply by 100 to get percentage used.
maxFacts	Long	Maximum number of facts allowed within the CEP server. The limit is soft and does not prevent publishing more facts.

Appendix E: Required Linux Platform Configurations

The following configurations are required on Linux platforms to ensure the meshIQ Platform runs smoothly and efficiently.

Table E-1. Linux Configurations

Description	Configuration
Newer glibc-2.2.x libraries cannot handle initial thread stack sizes greater than 6MB and can cause a segmentation fault. (Red Hat 7.0, Mandrake 8.0, SUSE 7.2, Debian 2.2)	Use "u-limit -s 2048" in bash shell or "limit stacksize 2048" in tcsh to limit the initial thread stack to 2MB.
A thread that is waiting on an I/O operation will not wake up if a file involved in the I/O operation is closed.	Set the J2SE_PREEMPTCLOSE environment variable to 1. J2SE_PJREEMPTCLOSE=1 export J2SE_PREEMPTCLOSE

Appendix F: Dashboard Database Schema

Table F-1. Historical Sensor Names		
Sensor Name	Data Type	Default
SID	integer	NOT NULL
SensorName	char (255)	NOT NULL
Severity	char (32)	NOT NULL
SensorValue	char (255)	NULL
NumericValue	float	NULL
Health	float	NULL
LogTime	datetime	NULL
EventID	integer	NULL
ServerName	char (255)	NULL
OSName	char (128)	NULL
OSVersion	char (12)	NULL
OSArch	char (12)	NULL
OS User	char (48)	NULL
APUser	char (48)	NULL
ServiceCategory	integer	NULL
ServiceType	char (48)	NULL
ObjectType	char (48)	NULL
Location	char (128)	NULL
SensorIndex	char (64)	NULL

Table F-2. Real-Time Sensor Names		
Sensor Name	Data Type	Default
SID	integer	NOT NULL
SensorName	char (255)	NOT NULL
Severity	char (32)	NOT NULL

Table F-2. Real-Time Sensor Names		
Sensor Name	Data Type	Default
SensorValue	char (255)	NULL
NumericValue	float	NULL
Health	float	NULL
LogTime	datetime	NULL
EventID	integer	NULL
ServerName	char (255)	NULL
OSName	char (128)	NULL
OSVersion	char (12)	NULL
OSArch	char (12)	NULL
OSUser	char (48)	NULL
APUser	char (48)	NULL
ServiceCategory	integer	NULL
ServiceType	char (48)	NULL
ObjectType	char (48)	NULL
Location	char (128)	NULL
SensorIndex	char (64)	NULL

Table F-3. Policy Names		
Policy Name	Data Type	Default
SID	integer	NOT NULL
ManagerName	char (128)	NOT NULL
PolicyName	char (128)	NOT NULL
DomainName	char (128)	NOT NULL
TableName	char (128)	NOT NULL
RssUrl	char (128)	NOT NULL
RssItemUrl	char (128)	NOT NULL
TwoorksUrl	char (128)	NOT NULL

Appendix G: Derived Metrics

Table H-1. Derived Metrics

Derived Metric	Description	Formula
Value	Current value of fact	
Name	Complete fact name (all tree levels, from root to leaf node)	
Class	Class of fact (e.g., java lang long)	
Length	Length of fact, in bytes	
Update-Count	Number of times the fact was updated (changed or same)	
Change-Count	Number of times the fact was changed	
Reset-Count	Number of times the fact was reset	
Previous-Value	Previous value of the fact	
Time-Created	Time the fact was created	
Last-Updated	Time of most recent update (changed or same)	
Last-Changed	Time of most recent change	
Update-Age	Time since last update	
Change-Age	Time since last change	
Time-Difference	Time difference in ms between fact publisher (origin) and subscriber	
Min	Overall minimum value since reset	
MAvg	Moving average	
Counter	Last actual value for a counter type, versus the delta reported	

Table H-1. Derived Metrics		
Derived Metric	Description	Formula
Ignore-Status	Ignore status indicator – if ignored, sensor alerting will not trigger	
Time-Since-Reset	Time since reset	
Change-Latency	Time between latest changes	
Update-Latency	Time between latest updates	
Update-Velocity	Rate of update	
History-Size	Number of facts in history store	
History-Locked	If the history is defined but locked, new numbers will not be put into history when they arrive	
History-Max-Size	Maximum number of history samples	
History-Time	Time of fact history	
History-Avg	Average of values in history facts	
History-EMAvg	Exponential Moving Average of values in fact history	=ema(last 10 samples)
History-Max	Maximum value in fact history	=max(all history samples)
History-Min	Minimum value in fact history	=min(all history samples)
History-Variance	Variance of values in fact history	=var(all history samples)
History-Deviation	Standard Deviation of values in fact history	=stdev(all history samples)
History-Dev-Mean	Number of standard deviations from the mean	current - mean(history))/stdev(history)
History-Bound	Upper bound based on Chebyshev in-equality	http://en.wikipedia.org/wiki/Chebyshev's_inequality

Table H-1. Derived Metrics

Derived Metric	Description	Formula
History-Band-High	High band based on Bollinger bands	http://en.wikipedia.org/wiki/Bollinger_bands
History-Band-Low	Low band based on Bollinger bands	http://en.wikipedia.org/wiki/Bollinger_bands
History-RSI	Relative Strength Indicator	http://en.wikipedia.org/wiki/Relative_strength_index
History-SO-K	Stochastic oscillator	http://en.wikipedia.org/wiki/Stochastic_oscillator
History-CAvg	Average percent change in history (based on % change)	
History-CVariance	Variance of values in fact history (based on % change)	
History-CDeviation	Standard Deviation of values in fact history (based on % change)	
History-CBound	Upper bound based on Chebyshev in-equality (based on % change)	
History-CDev-Mean	number of standard deviations from the mean (based on % change)	
History-CBand-High	High band based on Bollinger bands (based on % change)	
History-CBand-Low	Low band based on Bollinger bands (based on % change)	
History-CAvg-Gain	Average Percent Gain	
History-CAvg-Loss	Average Percent Loss	
History-CAD-Ratio	Ratio of Advances to Declines	# advances (increases) / # declines (decreases)
History-HROC	Historical rate of change percent	$(\text{current} - \text{old_history}) / \text{old_history}$
History-IROC	Instantaneous rate of change percent	$(\text{current} - \text{last}) / \text{last}$
Expiry-Time	Defined life of the fact in milliseconds. If the	

Table H-1. Derived Metrics		
Derived Metric	Description	Formula
	fact is not updated within this time, it will be deleted	
Expiry-Timer	The amount of time left before it expires (when 0, it will be deleted)	

Appendix H: Software Acknowledgements

I.1 Open-Source Components

Components are listed below according to their respective applicable licenses.

I.2 Open-Source Components licensed under the Apache license version 2.0

apache / commons, apache / httpcomponents-client, apache / httpcomponents-core, apache / logging-log, apache / logging-log4j2, apache / maven, apache / pdfbox, apache / xmlgraphics-batik, Apache 2.0 / activemq, Apache 2.0 / commons, Apache 2.0 / flume, Apache 2.0 / Hadoop, Apache 2.0 / logging-log, Apache 2.0 / maven, Apache 2.0 / poi, Apache 2.0 / zookeeper, Apache Tomcat Application Server, Apache Web Server, atlassian / httpclient org.Apache 2.0.httpcomponents:httpmime

Active MQ, bbqrob / merge-maven-plugin org.zcore.maven:merge-maven-plugin, bcaudan / jasmine-spec-reporter, brettwooldridge / HikariCP com.zaxxer:HikariCP, Codepage, com.github.logstash-plugins:logstash-input-beats, com.google.auto.service:auto-service, com.thetransactioncompany:java-property-utils, dropwizard / metrics, eirslett / frontend-maven-plugin com.github.eirslett:frontend-maven-plugin, emacsist / tomcat-source-maven org.apache.tomcat:tomcat-jdbc, facebookarchive / ini4j org.ini4j:ini4j, FasterXML / Jackson, google / gson com.google.code.gson:gson, google / guava com.google.guava:guava, google / re2j com.google.re2j:re2j, iclouding / kafka-src-

Infinispan, jettison-json / jettison org.codehaus.jettison:jettison, JODA Time, JSON-Simple, JUG, Kafka, Log4jV1/2, LuisSala / CORS-Java-Servlet-Filter com.thetransactioncompany:cors-filter, mojohaus / jaxb2-maven-plugin, mojohaus / versions-maven-plugin, net.java.dev.jna:jna-platform, netty / netty io.netty:netty-all

OpenAPITools / openapi-generator, OpenHFT / Chronicle-Queue, org.codehaus.groovy:groovy-jsr223, org.junit.platform:junit-platform-surefire-provider, org.junit.platform:maven-surefire-plugin, org.mockito:mockito-junit-jupiter, org.sonatype.plugins:nexus-staging-maven-plugin, palantir / tslint, puppeteer / puppeteer, qos-ch / logback ch.qos.logback:logback-classic, quartz-scheduler / quartz org.quartz-scheduler:quartz,

rabbitmq / rabbitmq-jms-client, RbkGh / Jose4j org.bitbucket.b_c:jose4j, ReactiveX / rxjs, ReactiveX / rxjs-tslint, request / request, rholder / fluent-hc-, robtimus / sftp-fs com.github.robtimus:sftp-fs, rxjs, rxjs-compat, schibsted / jslt com.schibsted.spt.data:jslt, SheetJS / js-codepage codepage, Solr Cloud, STORM, tslib, web-animations / web-animations-js, WSO2 Siddhi CEP, xmlunit / xmlunit org.xmlunit:xmlunit-matchers, yadickson / jsonpath-debs com.jayway.jsonpath:json-path, zhouliang3 / DA2:A117-maven com.lmax:disruptor

I.2.1 Terms of the Apache license version 2.0

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"**Licensor**" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"**Legal Entity**" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "**control**" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"**You**" (or "**Your**") shall mean an individual or Legal Entity exercising permissions granted by this License.

"**Source**" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"**Object**" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"**Work**" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"**Derivative Works**" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"**Contribution**" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "**submitted**" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "**Not a Contribution**."

"**Contributor**" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "**NOTICE**" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

I.3 Open-Source Components licensed under the BSD 3-Clause

ANTLR

Copyright (c) 2012 Terence Parr and Sam Harwell
All rights reserved.

d3-collection

Copyright 2010-2016, Mike Bostock

d3-ease

Copyright 2010-2021 Mike Bostock
Copyright 2001 Robert Penner

is / jsch com.jcraft:jsch

Copyright (c) 2002-2015 Atsuhiko Yamanaka, JCraft, Inc.
All rights reserved.

I.3.1 Terms of the BSD 3-Clause

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JCraft, Inc. (Modified BSD 3-clause license)

JSch 0.0.* was released under the GNU LGPL license. Later, we have switched over to a BSD-style license.

Copyright (c) 2002-2015 Atsuhiko Yamanaka, JCraft, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED `AS IS' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE

OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

I.4 Eclipse Distribution License v.1.0

com.sun.xml.bind:jaxb-core, jakarta.xml.ws:jakarta.xml.ws-api, org.glassfish.jaxb:jaxb-runtime,

I.4.1 Terms of the EDL v1.0 license

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

I.5 Open-Source Components licensed under the EPL 1.0 (Eclipse Public License 1.0)

junit-team / junit4 junit:junit, org.eclipse.jetty.websocket:websocket-client and eclipse / jetty.project org.eclipse.jetty:jetty-client (SU33 and earlier).

I.5.1 Terms of the EPL 1.0

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial

product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

1.6 Open-Source Components licensed under the EPL 2.0 (Eclipse Public License 2.0)

eclipse / jetty.project org.eclipse.jetty:jetty-client and org.eclipse.jetty.websocket:websocket-client (SU34 and later), eclipse / paho.mqtt.java org.eclipse.paho:org.eclipse.paho.client.mqttv3, eclipse-ee4j / common-annotations-api jakarta.annotation:jakarta.annotation-api, jakartaee / messaging jakarta.jms:jakarta.jms-api, junit-team / junit5 org.junit.jupiter:junit-jupiter-engine, org.eclipse.m2e:lifecycle-mapping,

org.glassfish.jersey.containers:jersey-container-servlet, org.glassfish.jersey.inject:jersey-hk2,
org.glassfish.jersey.media:jersey-media-json-jackson, org.glassfish.jersey.media:jersey-media-multipart

I.6.1 Terms of the EPL 2.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial content Distributed under this Agreement, and
- b) in the case of each subsequent Contributor:
 - i) changes to the Program, and
 - ii) additions to the Program;

where such changes and/or additions to the Program originate from and are Distributed by that particular Contributor. A Contribution "originates" from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include changes or additions to the Program that are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.
- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.
- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.
- e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

- a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and
- b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:

- i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
- ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
- iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and
- iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

- a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and
- b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ('notices') contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance

claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this

Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

I.7 Open-Source Components licensed under the ISC license

d3-brush, d3-dispatch, d3-drag, d3-force, d3-format, d3-hierarchy, d3-interpolate, d3-quadtree, d3-scale, d3-selection, d3-shape, d3-time, d3-time-format, d3-timer, d3-transition

Copyright 2010-2021 Mike Bostock

d3-array, d3-color

Copyright 2010-2022 Mike Bostock

d3-path

Copyright 2015-2021 Mike Bostock

I.7.1 Terms of the ISC license

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

I.8 Open-Source Components licensed under the LGPL

Java Native Access (JNA)

mariadb-corporation / mariadb-connector-j org.mariadb.jdbc:mariadb-java-client

Logback

Copyright (C) 1999-2017, QOS.ch. All rights reserved.

I.8.1 Terms of the LGPL

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish

on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a

"work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,

INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

I.9 Open-Source Components licensed under the MIT license

actions / checkout actions/checkout, actions / create-release actions/create-release, actions / setup-java actions/setup-java

Copyright (c) 2018 GitHub, Inc. and contributors

ajv-validator / ajv

Copyright (c) 2015-2021 Evgeny Poberezkin

andyearnshaw / Intl.js intl

Copyright (c) 2013 Andy Earnshaw

angular / angular

Copyright (c) 2010-2022 Google LLC. <http://angular.io/license>

angulartics / Angulartics

Copyright (c) 2021 angulartics

atoa

Copyright © 2015 Nicolas Bevacqua

base64-js, beatgammit / base64-js

Copyright (c) 2014 Jameson Little

Bootstrap

Copyright (c) 2011-2022 Twitter, Inc.

Copyright (c) 2011-2022 The Bootstrap Authors

Brace

Copyright 2013 Thorsten Lorenz.

carlos8f / node-idgen idgen

chrisbottin / xml-formatter

Copyright 2019 Chris Bottin (<https://github.com/chrisbottin>)

CirclonGroup / angular-tree-component

Copyright (c) 2020 Circlon Group

crossvent

Copyright © 2014 Nicolas Bevacqua

custom-event

Copyright (c) 2015 Nathan Rajlich

DefinitelyTyped / DefinitelyTyped

Copyrights are respective of each contributor listed at the beginning of each definition file.

dfederm / karma-jasmine-html-reporter

Copyright (C) 2011-2013 Vojta Jína and contributors.

Dragula

Copyright © 2015-2016 Nicolas Bevacqua

eligrey / FileSaver.js file-saver

Copyright © 2016 Eli Grey.

emitter-component

Copyright (c) 2014 Component contributors <dev@component.io>

eslint / eslint

Copyright OpenJS Foundation and other contributors, <www.openjsf.org>

floating-ui / floating-ui popper.js

Copyright (c) 2021 Floating UI contributors

isaacplmann / ngx-contextmenu

jasmine / jasmine jasmine-core

Copyright (c) 2008-2019 Pivotal Labs

<https://github.com/jasmine/jasmine/blob/main/MIT.LICENSE>

jQuery

Copyright OpenJS Foundation and other contributors, <https://openjsf.org/>

Jsczip

Copyright (c) 2009-2016 Stuart Knightley, David Duponchel, Franz Buchinger, António Afonso

juliangruber / stream

Copyright (c) 2012 Julian Gruber <julian@juliangruber.com>

justmoon / node-extend extend

Copyright (c) 2014 Stefan Thomas

karma-runner / karma

Copyright (C) 2011-2021 Google, Inc.

Lie

lifaon74 / url-polyfill

Copyright (c) 2017 Valentin Richard

lodash / lodash

Copyright JS Foundation and other contributors <<https://js.foundation/>>

lucastheisen / jsch-nio.com.pastdev:jsch-nio

Copyright (c) 2018 Lucas Theisen

mattlewis92 / angular-resizable-element

Copyright (c) 2016 Matt Lewis

mattlewis92 / karma-coverage-istanbul-reporter

Copyright (c) 2017 Matt Lewis

metcalfc / changelog-generator metcalfc/changelog-generator

Copyright (c) 2020 Chad Metcalf

mgechev / codelyzer

Copyright (c) 2016 Minko Gechev

mobxjs / mobx

Copyright (c) 2015 Michel Weststrate

mockito / mockito org.mockito:mockito-core

Copyright (c) 2007 Mockito contributors

ng2-ace-editor

Copyright (c) 2016 Kyle J. Kemp

ng2-dragula, MIT

Copyright (c) 2015-2016 Dmitriy Shekhovtsov <valorkin@gmail.com>

Copyright (c) 2015-2016 Valor Software

Copyright (c) 2015-2016 Nathan Walker <NathanWalker>

ng-bootstrap / ng-bootstrap

Copyright (c) 2015-2018 Angular ng-bootstrap team

ng-packagr / ng-packagr

Copyright (c) 2017-2020 Alan Agius

ng-select/ng-select

Copyright (c) Andžej Maciusovič

ngx-contextmenu

ngx-toastr

qos-ch / slf4j

Copyright (c) 2004-2022 QOS.ch Sarl (Switzerland)

sass / dart-sass sass

Copyright (c) 2016, Google Inc.

scttcper / ngx-toastr

Copyright (c) Scott Cooper <scttcper@gmail.com>

SLF4J

Copyright (c) 2004-2017 QOS.ch at <https://www.slf4j.org/license.html>

Stuk / jszip

Copyright (c) 2009-2016 Stuart Knightley, David Duponchel, Franz Buchinger, António Afonso

substack / node-falafel falafel

Copyright (c) 2012 James Halliday

swimlane/ngx-charts

Copyright (c) 2017 Swimlane

swimlane/ngx-datatable

Copyright (c) 2019 Swimlane <info@swimlane.com>

thlorenz / brace

Copyright 2013 Thorsten Lorenz.

ticky

Copyright (c) 2017 Guodong Xu

twbs / bootstrap

Copyright (c) 2011-2022 Twitter, Inc.

Copyright (c) 2011-2022 The Bootstrap Authors

TypeStrong / ts-node

Copyright (c) 2014 Blake Embrey (hello@blakeembrey.com)

valor-software / ng2-dragula

Copyright (c) 2015-2016 Dmitriy Shekhovtsov <valorkin@gmail.com>

Copyright (c) 2015-2016 Valor Software

Copyright (c) 2015-2016 Nathan Walker <NathanWalker>

ValYouW / eslint-plugin-named-unassigned-functions

ValYouW / njsTrace njstrace

Webpack

Copyright JS Foundation and other contributors

zefoy / ngx-color-picker

Copyright (c) 2016 Zef Oy

zloirock / core-js

Copyright (c) 2014-2022 Denis Pushkarev

zone.js

Copyright (c) 2016-2018 Google, Inc.

1.9.1 Terms of the MIT license

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

I.10 Open-Source Components licensed under the OFL 1.1

font-awesome

I.10.1 Terms of the OFL 1.1

SIL OPEN FONT LICENSE

Version 1.1 - 26 February 2007

PREAMBLE

The goals of the Open Font License (OFL) are to stimulate worldwide development of collaborative font projects, to support the font creation efforts of academic and linguistic communities, and to provide a free and open framework in which fonts may be shared and improved in partnership with others.

The OFL allows the licensed fonts to be used, studied, modified and redistributed freely as long as they are not sold by themselves. The fonts, including any derivative works, can be bundled, embedded, redistributed and/or sold with any software provided that any reserved names are not used by derivative works. The fonts and derivatives, however, cannot be released under any other type of license. The requirement for fonts to remain under this license does not apply to any document created using the fonts or their derivatives.

DEFINITIONS

"Font Software" refers to the set of files released by the Copyright Holder(s) under this license and clearly marked as such. This may include source files, build scripts and documentation.

"Reserved Font Name" refers to any names specified as such after the copyright statement(s).

"Original Version" refers to the collection of Font Software components as distributed by the Copyright Holder(s).

"Modified Version" refers to any derivative made by adding to, deleting, or substituting — in part or in whole — any of the components of the Original Version, by changing formats or by porting the Font Software to a new environment.

"Author" refers to any designer, engineer, programmer, technical writer or other person who contributed to the Font Software.

PERMISSION & CONDITIONS

Permission is hereby granted, free of charge, to any person obtaining a copy of the Font Software, to use, study, copy, merge, embed, modify, redistribute, and sell modified and unmodified copies of the Font Software, subject to the following conditions:

- 1) Neither the Font Software nor any of its individual components, in Original or Modified Versions, may be sold by itself.
- 2) Original or Modified Versions of the Font Software may be bundled, redistributed and/or sold with any software, provided that each copy contains the above copyright notice and this license. These can be included either as stand-alone text files, human-readable headers or in the appropriate machine-readable metadata fields within text or binary files as long as those fields can be easily viewed by the user.
- 3) No Modified Version of the Font Software may use the Reserved Font Name(s) unless explicit written permission is granted by the corresponding Copyright Holder. This restriction only applies to the primary font name as presented to the users.
- 4) The name(s) of the Copyright Holder(s) or the Author(s) of the Font Software shall not be used to promote, endorse or advertise any Modified Version, except to acknowledge the contribution(s) of the Copyright Holder(s) and the Author(s) or with their explicit written permission.
- 5) The Font Software, modified or unmodified, in part or in whole, must be distributed entirely under this license, and must not be distributed under any other license. The requirement for fonts to remain under this license does not apply to any document created using the Font Software.

TERMINATION

This license becomes null and void if any of the above conditions are not met.

DISCLAIMER

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

I.11 Open-Source Components licensed under the Zlib license

```
/* zlib.h -- interface of the 'zlib' general purpose compression library
   version 1.2.12, March 27th, 2022
   Copyright (C) 1995-2022 Jean-loup Gailly and Mark Adler
```

I.11.1 Terms of the Zlib license

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Index

%	
%date%	156, 169
%desc%	156, 169, 173
%event%	156, 169, 173
%facts%	156, 169, 173
%from%	156, 169
%id%	156, 169
%ivalue%	156
%party%	156, 169
%root%	156, 169
%sev%	156, 170, 173
%time%	156, 170
%user%	156, 170
.	
.EVT	210, 248
.pmv	220
A	
Action Manager	183, 184
action parameters	172
ACTION_ACTIVITY	210, 248
adding groups	63
adding new users	60
AIM	75
alert options	145, 158, 171
alert setup	145
alerts	143
apnet	39, 264
arming policies	125
ATPNames	32, 33, 37, 235, 268
ATPNode	33, 235, 237, 267, 268
automating sensor actions	172
B	
Built-in Policies	121
business view	136
Business View Event Logs	198
Business View Explorer	137
business view file	125
Business View History	212
Business View Sensor Overview	205
C	
car expert	76
changing alert properties	143
chart options	203
chart profiles	220
charting	219
Command Prompt, start from	32
console profile	62
contacts	106
Contacts	112
counter expert	76
D	
deleting user accounts	65
Dependencies	112
Deploy Manager	109
Deploy Policies	120
deploying file monitor	98
deploying process wrapper	83
deployment cycle, business view	136
displaying sensors graphically	181
Documentation Library	47
E	
Email	111
E-mail Notification, Manager	112
email notifications	145, 158, 171
Enable Paging	113, 116
event filter	216
event log options	215
event log properties	211
event logging properties	192, 218
event logs	212, 213
event viewer	211
event, find	217
experts	75
Experts	
built-in	76
samples	75
Exporting Log Files	214
F	
Fact volatility	141
facts	106
Feedback, User	2
File Monitor	98
filtering events	216
find events	217

G

graphically displaying sensors 180

group names.....63

group removal65

H

Health tab..... 185

History, Business View 212

I

ignoring sensor actions 255

K

key performance properties 247

Keystore..... 20, 25

Keytool26

L

LAX Customization 237

LDAP.....19

Load Business View Policy..... 119

Log Files, Saving..... 214

log settings 218

logging 113, 210

logging characteristics 212

logging properties,sensor..... 190

logs, event 212

M

Mail Server

 Outgoing 112

Main Toolbar40

maintenance schedule

 ignore schedule..... 187

 operational schedule..... 188

Management Console.....40

Manager

 built-in 107

 deploy..... 109

 Policy 107

 Speed..... 107

Manager Configuration..... 109

manager properties 111

managers..... 106

Managing Sensors..... 175

Menus40

monitor, performance 220

N

naming.xml 236

New Filter Policy 119

New Schedule Policy 119

new users60

Node, stop.....36

node.properties..... 238

notification 145, 158, 171

Notify Delay..... 145, 158, 171

NRD file..... 253

O

operational properties..... 238

options, apnet..... 264

options, event log..... 215

Outgoing Mail Server 112

P

Paging 113, 116

performance monitor 219

Performance Properties 235

Performance Tool..... 219

Platform Dependencies 112, 125

Policies

 built-in 121

Policies 119

 deploy..... 120

POLICIES 248

Policy

 Load Business View 119

 New Filter 119

 New Schedule..... 119

policy, configure 125

Process Modeler 179

process wrapper

 deploy..... 79, 83

Process Wrapper

 Configure84

profiles, charting..... 220

properties, key performance 247

properties, manager 111

properties, operational..... 238

properties, Search Tool..... 223

properties,operational..... 235

properties,performance 235

Pull-Down Menu40

R

README.TXT2

Re-Arm Delay 145, 158, 171

recipient, E-mail..... 112

record facts 113
 Recording tab.....91
 registry.xml 236
 related views 208
 remove users65
 removing groups65

S

sample experts76
 Saving Log Files..... 214
 Scheme 145, 158, 171
 Search Tool..... 223
 security.dat..... 236
 SECURITY_ACTIVITY 210, 248
 sensor displaying, graphically 180
 sensor event logging properties..... 190
 sensor settings, ignore..... 255
 Server Registry Files 236
 Server,Paging 113, 114, 116
 Service Dependencies..... 112, 125
 service name, policies 125
 SERVICES..... 248
 severities..... 143
 sorting event log..... 215
 Speed Manager 107
 SQL Query79
 start AutoPilot, UNIX37
 Starting AutoPilot/IT32
 Starting Core Services30

state change delay 185
 statistics policies..... 125
 Stopping AutoPilot/IT36
 Stopping Node36
 streaming options 229
 Stress Test78
 SYSTEM 7, 210, 218, 248, 264

T

technical support.....2
 Toolbar.....40

U

UNNAMED 210, 218, 248, 249
 URL Monitor76
 user actions..... 145, 158, 171, 172
 User Authentication 31, 34
 User Feedback2
 user group removal.....65
 User Name
 E-mail..... 112
 user password changes.....66

V

Version Control62
 viewing logs..... 218
 Volatility 141